

DISCOVERY

IN

DIGITAL

TRACES

CENTRAL FISH INTEREST

DAVID GRAUS

Entities of Interest

Discovery in Digital Traces

David Graus

Entities of Interest

Discovery in Digital Traces

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. K.I.J. Maex
ten overstaan van een door het College voor Promoties ingestelde
commissie, in het openbaar te verdedigen in
de Agnietenkapel
op vrijdag 16 juni 2017, te 10:00 uur

door

David Paul Graus

geboren te Beuningen

Promotiecommissie

Promotor:

prof. dr. M. de Rijke

Universiteit van Amsterdam

Co-promotor:

dr. E. Tsagkias

904Labs

Overige leden:

prof. dr. A.P.J. van den Bosch

Radboud Universiteit Nijmegen

prof. dr. ing. Z.J.M.H. Geradts

Universiteit van Amsterdam

dr. ir. J. Kamps

Universiteit van Amsterdam

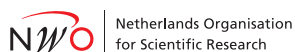
dr. E. Kanoulas

Universiteit van Amsterdam

prof. dr. D.W. Oard

University of Maryland

Faculteit der Natuurwetenschappen, Wiskunde en Informatica



The research was supported by the Netherlands Organisation for Scientific Research (NWO) under project number 727.011.005.



SIKS Dissertation Series No. 2017-23

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



The printing of this thesis was financially supported by the Co van Ledden Hulsebosch Center, Amsterdam Center for Forensic Science and Medicine.

Copyright © 2017 David Graus, Amsterdam, The Netherlands

Cover by Rutger de Vries/perongeluk.com

Printed by Off Page, Amsterdam

ISBN: 978-94-6182-800-2

Acknowledgements

I started the process that culminated in the little booklet you are now reading in 2012. At that time, I had never touched a unix, I didn't know what an ssh was, and I hardly knew how to write a line of code. I hate to be overly dramatic,¹ but I have grown, not only as a researcher, also as a person, and this growth is not exclusively to be attributed to the passing of time. Over the course of my PhD I have become more confident, about myself and my skills, and I have learned I can pick up anything I put my mind (and time) to. The latter is by far the greatest life's lesson my PhD has brought me.

For all of this, I have to thank some people. First, P&C, for setting up the prior by raising me in a home where I naturally got exposed to computers, technology, and (overly) critical thinking. P, for putting me and Mark in front of a Sinclair ZX81 at young age, traumatizing us with the scariest ASCII-rendered games imaginable. It shaped me into the tough trooper I am today. C, for putting up with a grumpy household when our attempts at assembling computers from parts scrambled from all over the place invariably failed.

Next, Maarten, for taking a "chance" with me. I was quite wet behind the ears, as one would say, before I started my PhD. I am glad and thankful that this did not hold Maarten back for giving me the opportunity to see how I would pan out as a computer scientist. I was a coin-flip. Thanks for flipping!

Next next, ILPS. First, the senior population at ILPS. In particular, Edgar, who at the time understood better than me what I was doing, and who saw that I could do and might enjoy this PhD. Manos and Wouter, who guided, supervised, co-authored, and co-lived this period with me. My peers at ILPS, of which I will name a few because these are my acknowledgements and I get to decide whether I single people out or not: Daan, Anne, Tom, and Marlies. I enjoyed interfering with your day-to-day activities, sharing misery, disagreeing (hi Anne), coffee breaks, bootcamps and runs, and everything else.

Then, my co-authors, for partaking in that thing we do in science, which conveniently provided the basis of this dissertation. Paul, Ryen, and Eric for mentoring me during my internship at Microsoft Research in the summer of 2015. Best summer ever.

Finally, my brothers from other mothers, Rutger and Marijn, for putting up with me being right all the time, slightly cocky, and (at times) tending towards the far-right end of the spectrum of self-confidence. I'm sorry.

Moving on to the nonliving things: Polder, Oerknal, Maslow, and Joost (in order of appearance) for providing shelter for unwinding, escaping, brainstorming, philosophizing and reflecting on life, academia, and beyond. Oh, and also for drinking beers. Twitter, for giving me a place where I could write down stuff when Christophe would stop listening to the monologues I typically directed at my computer screen (but were intended for all to hear). Spotify and Bose, for sheltering me from the real world during extended periods of (coding) cocooning.

Okay, this is not an exhaustive list, but I have to both start and stop somewhere. Before we part ways and move onto the researchy stuff, I will leave you with one big fat cliché—which I know loses some of its expressive power by virtue of being a cliché (but which I hope to counteract a bit by being explicit about my awareness of it being a cliché):

It was a great journey!

David

¹I really don't...

Contents

1	Introduction	1
1.1	Research Outline and Questions	3
1.2	Main Contributions	9
1.3	Thesis Overview	10
1.4	Origins	11
2	Background	13
2.1	Information Retrieval	13
2.1.1	Retrieval Models	14
2.1.2	Evaluation	16
2.2	Semantic Search	17
2.2.1	Knowledge Bases and Entities	17
2.3	Emerging Entities	20
2.3.1	Analyzing Emerging Entities	21
2.3.2	Predicting Emerging Entities	21
2.3.3	Constructing Entity Representations for Effective Retrieval	22
2.4	Analyzing and Predicting Activity	24
2.4.1	Mining Email	24
2.4.2	User Interaction Logs	25
2.5	What's next	27
I	Analyzing, Predicting, and Retrieving Emerging Entities	29
3	Analyzing Emerging Entities	31
3.1	Introduction	31
3.2	Research Questions	33
3.3	Data	34
3.3.1	Emerging Entities in News and Social Media	35
3.3.2	Time Series of Emerging Entity Mentions	36
3.4	Methods	37
3.4.1	Time Series Clustering	38
3.4.2	Time Series Analysis	40
3.5	Results	41
3.5.1	Emergence Patterns	41
3.5.2	Entity Emergence Patterns in Social Media and News	49
3.5.3	Emergence Patterns of Different Entity Types	53
3.6	Conclusion	57
3.6.1	Findings	58
3.6.2	Implications	59
3.6.3	Limitations	60
3.6.4	What's Next?	61

4	Predicting Emerging Entities	63
4.1	Introduction	63
4.2	Approach	65
4.3	Unsupervised Generation of Pseudo-ground Truth	65
4.4	Sampling Pseudo-ground Truth	66
4.4.1	Sampling Based on Entity Linker’s Confidence Score	66
4.4.2	Sampling Based on Textual Quality	67
4.5	Experimental Setup	68
4.5.1	Dataset	69
4.5.2	Experiment I: Sampling Pseudo-ground Truth	69
4.5.3	Experiment II: Prediction Experiments	69
4.5.4	Evaluation	69
4.6	Results	70
4.6.1	Experiment I: Sampling Pseudo-ground Truth	70
4.6.2	Experiment II: Impact of Prior Knowledge	72
4.7	Conclusion	74
5	Retrieving Emerging Entities	77
5.1	Introduction	77
5.2	Dynamic Collective Entity Representations	80
5.2.1	Problem Statement	80
5.2.2	Approach	80
5.2.3	Description Sources	81
5.2.4	Adaptive Entity Ranking	84
5.3	Model	84
5.3.1	Entity Representation	84
5.3.2	Features	85
5.3.3	Machine Learning	86
5.4	Experimental setup	87
5.4.1	Experiments	87
5.4.2	Baselines	89
5.4.3	Data Alignment	89
5.4.4	Machine Learning	90
5.4.5	Evaluation	90
5.5	Results and Analysis	91
5.5.1	Dynamic Collective Entity Representations	91
5.5.2	Modeling Field Importance	94
5.5.3	Ranker Adaptivity	95
5.6	Conclusion	97
II	Analyzing and Predicting Activity from Digital Traces	101
6	Analyzing and Predicting Email Communication	103
6.1	Introduction	103
6.2	Communication Graph	104

6.3	Modeling	105
6.3.1	Email Likelihood	105
6.3.2	Sender Likelihood	106
6.3.3	Recipient Likelihood	107
6.4	Experimental Setup	107
6.4.1	Evaluation	108
6.4.2	Parameter Tuning	108
6.5	Results and Analysis	108
6.5.1	Between Groups Comparison	109
6.5.2	Performance over Time	110
6.5.3	Rankers' Correlation	111
6.6	Conclusion	112
7	Analyzing and Predicting Task Reminders	115
7.1	Reminder Types	116
7.1.1	Reminder Composition	116
7.1.2	Data	117
7.1.3	Identifying Common Tasks	118
7.2	Task Type Taxonomy	119
7.3	Reminder Patterns	124
7.3.1	Method	124
7.3.2	Global Patterns	125
7.3.3	Temporal Patterns of Different Task Types	126
7.3.4	Temporal Patterns of Terms in Reminder Task Descriptions	129
7.3.5	Temporal Patterns of Notification Times	130
7.4	Predicting Notification Time	131
7.4.1	Experimental Setup	132
7.4.2	Results	134
7.5	Conclusion	135
8	Conclusions	139
8.1	Main Findings, Limitations, and Implications	140
8.2	Future Research Directions	144
	Bibliography	147
	Samenvatting	173
	Summary	174

1

Introduction

“Dis-moi ce que tu manges, je te dirai ce que tu es.”

—Jean Anthelme Brillat-Savarin

From the things you “like” on Facebook, algorithms can infer many personal and demographic traits with surprisingly high accuracy. For example, political preferences, religious beliefs, but even more obscure factoids, such as whether someone’s parents were together during their childhood [276]. The predictive power of these algorithms relies on the availability of *digital traces* of a large number of people. In the era of *Big Data*, digital traces are available in abundance. We willingly share, post, like, link, play, and query, all the while leaving behind rich digital traces. Traces that, when aggregated, can provide meaningful insights into behavior, preferences, and profiles of people.

We distinguish between two types of digital traces: *Active digital traces* that people leave behind deliberately, e.g., blogposts, tweets, or forum posts, and *passive digital traces*, that people may leave behind unknowingly, e.g., by clicking on links on websites, querying web search engines, or by simply visiting websites [94]. The quality of and our reliance on online services has enabled wide availability of these digital traces. Combined with rapid developments in computer hardware, software, and algorithms, the availability of our digital traces has given rise to (yet) a(nother) resurgence of Artificial Intelligence (AI). With it, a new economy is on the rise that revolves around applied *machine learning* for understanding users, e.g., inferring their traits, preferences, and behavior, for applications such as targeted advertising, personalizing content, and more generally, improving online services.

People have voiced concerns around this rise of the “algorithm,” hailing it as the demise of our privacy and freedom [242], the death of politics [189], and in general harmful [247]. But indisputably, mining, collecting, analyzing, and leveraging our digital traces has brought many advances. At its most visible level, it has permeated into our day-to-day lives, and has enabled our access to exceeding and unforeseen amounts of data, information, and knowledge. While this information access may seem self-evident to us, it is not. Enabling effective access to these huge amounts of information relies to a large degree on understanding people’s behavior, preferences, and traits. The “relevance” of information is highly user-specific, it depends on the context of the user (e.g., the time of day, geographical location, or a task the user may be executing), and more personal aspects such as taste, preference, or background [186].

We would be unable to find what we wanted if it were not for the search engines that learn from their users [122, 227], the filtering algorithms that find and prioritize content relevant to us [54, 77], and the recommendation algorithms that allow services such as Netflix and Spotify to serve a wider variety of content, by no longer being constrained to catering to the masses like traditional media.

Next to these everyday-life, practical applications, analyzing and mining digital traces has brought advances for (social) good, too. In the health domain, search engine logs have led to discoveries of new and previously unreported side effects of prescription drugs [261], and adverse drug reactions [275]. Social media posts have allowed us to develop methods for identifying people who suffer from depression [56]. And more generally, the research fields of sociology, anthropology, (social) psychology, and (digital) humanities have benefited from the possibilities brought by large-scale analysis of (digital) traces [166].

And this is just the beginning. More and more areas are starting to reap the benefits of advances in artificial intelligence and the ability to extract information from large collections of digital data.

Discovery in Digital Traces

The Enron scandal [264], the Hillary Clinton email controversy [263], the Panama papers [265], or any data-release by WikiLeaks [266], are all examples of cases in which large amounts of (digital) traces needed to be investigated, explored, and turned upside down, to gain insights and discover “evidence.” This *E-Discovery* task is at its core “*finding evidence of activity in the real world*” in “Electronically Stored Information” (ESI) [197]. Discovery in digital data finds applications in many domains and scenarios, e.g., in (investigative) journalism [151], digital forensics [128, 167], and litigation, where data may be requested by a plaintiff with the aim of gaining insights and finding evidence for a legal case [197].

Whether in litigation, journalism, or digital forensics, in this discovery task, users set out to browse and explore large collections of data (usually text), with the aim of discovering new information, or of finding answers to any questions they may have.

Gaining insights and making sense of our digital traces may seem trivial. Often in written form, digital traces hold the promise of being explicit, structured, unambiguous, and readily interpretable. However, even written language is not as structured as it may seem. First, language is noisy. Ambiguity is common across languages, the meaning of words may change over time, and without understanding the continuously changing context in which language is created, it is often impossible to understand its intended meaning [87]. Second, the amount of digital data we produce on a daily basis is enormous and ever-growing. We are exposed to and produce an ever-increasing volume of data, both online on e.g., social media, collaboration platforms, and email, and offline on e.g., laptops, mobile phones, computers and storage media [161]. To address the above challenges, the need arises for automated methods for *sense-making* of digital traces for discovery, with the goal of both understanding their *content* and the *contexts* in which they are created.

In this thesis, we draw inspiration from this discovery scenario, and propose methods that aim to support sense-making of digital traces. We focus on textual traces, e.g., emails,

social media streams, and user interaction logs. In two parts in this thesis, we address two different aspects that are central to sense-making from digital traces.

In the first part of the thesis, we study methods for sense-making of the *content* of (textual) digital traces. More specifically, our objects of study are the *real-world entities* that are referenced in digital traces. Knowing, e.g., which people are mentioned in email communication between employees of a company, which companies are discussed by stockbrokers on Twitter, or which people and companies are mentioned in the Panama papers, is central to the exploratory and investigative search process that is inherent to sense-making [268]. In this part, our *entities of interest* are real-world entities: “things with distinct and independent existence” [1] that exist *in the real world*. We thus study the occurrence of concrete entities, e.g., companies, organizations, locations, and products in digital traces such as email, social media, and blogs.

In the second part of the thesis, we focus on the *context* of digital traces. Here, our *entities of interest* are the producers of digital traces, i.e., the people that leave behind the traces. Uncovering real-world activity is a central task in the E-Discovery sense-making process [197], thus, we focus on methods for uncovering this (evidence of) real-world activity from digital traces. Our aim is to understand real-world behavior from digital traces. We present two case studies where we analyze and leverage patterns in behavior to predict activity of the people who leave behind digital traces.

1.1 Research Outline and Questions

As outlined above, we distinguish two research themes on automated methods for sense-making of digital traces:

In the first part, our *entities of interest* are the real-world entities that are referenced in digital traces. Identifying the real-world entities that appear in ESI such as emails, social media, or forums and blogs, supports the exploratory and complex search process that underlies the discovery process. In the discovery scenario, the *entities of interest* may not be known in advance, i.e., they may not be well-known or established entities. For this reason, we focus on so-called *emerging* real-world entities, i.e., entities that are not described (yet) in publicly available knowledge bases such as Wikipedia. We address three tasks, (i) analyzing, (ii) predicting, and (iii) retrieving newly emerging real-world entities.

In the second part of this thesis, our *entities of interest* are the producers of the digital traces. We aim to (i) understand their behavior and real-world activity by analyzing digital traces, and (ii) predict their (future) real-world activity, leveraging the analyses and uncovered patterns. In this part, we present two case studies where we (i) analyze and predict email communication by studying the communication network and email content, and (ii) analyze interaction data with a mobile intelligent assistant to predict when a user will perform an activity.

Part I. Analyzing, Predicting, and Retrieving Emerging Entities

The discovery task is a lot like looking for needles in haystacks, without knowing what the needles look like. Users of search systems in the discovery context (e.g., forensic

analysts, lawyers, or journalists) may be interested in finding evidence of real-world activity, without knowing beforehand what type of activity nor whose activity they are looking for. Traditional Information Retrieval (IR) methods fall short in supporting this type of exploratory search [237]. They largely rely on *lexical matching*, i.e., matching words in a search query to words in a document, to allow users to retrieve documents from a collection. However, this type of search is not suitable for the typical discovery scenario, where the aim is to understand the 5 W's: *Who was involved? What happened? Where, when and why did it happen?* [197]. For answering these questions, it is typically difficult to formulate an exhaustive set of search queries, which means the keyword-based lexical matching paradigm does not suffice.

Semantic Search, i.e., “search with meaning” [20], is an alternate search paradigm that aims to move beyond lexical matching, and improve document retrieval by incorporating additional (external) knowledge in the search process. This additional knowledge can be in the form of the discussion structure of email threads, the topical structure of documents, or information on the entities and their relations that are mentioned in documents [250].

In the first part of this thesis we focus on real-world entities that appear, i.e., are referenced, in digital traces. Understanding which real-world entities appear and are discussed in digital traces is of central importance in exploratory search [140], and in answering the 5 W's.

A central task in semantic search is Entity Linking (EL) [20]. EL revolves around linking the mentions of real-world entities in text to their representations in an external Knowledge Base (KB). Linking mentions of real-world entities in text to their KB representations effectively boils down to disambiguation, and enables the enrichment of text with additional information and structure. Knowing the unambiguous real-world entities that occur in text enables better support for searching through large text collections [26] and the complex information seeking behavior that is inherent to the discovery process [268], through, e.g., enabling filtering for specific entities (e.g., a person) or entity types (e.g., companies), and by providing insights into how different entities co-occur.

EL relies on external knowledge from a reference KB that contains (descriptions of) real-world entities [216]. An example and often-used instantiation of such a reference KB is Wikipedia, the world's largest encyclopedia. Wikipedia is online, collectively built, and known to democratize information [256]. At the time of writing, the English Wikipedia contains over 5 million articles that represent both abstract and real-world entities, written by over 29 million users [267]. With this broad coverage, it spans the majority of entities that emerge in the media, and play a role in our public discourse, it has been dubbed a “global memory place,” where collective memories are built [205]. In addition, the rich metadata of Wikipedia, e.g., its hyperlink structure, category structure, and infoboxes, can be effectively employed as additional signals for disambiguating and improving EL, which makes Wikipedia an attractive standard reference KB for EL systems.

Due to the ambiguous nature of language, knowing the words that refer to entities (i.e., *entity mentions*) alone is unlikely to be sufficient for answering the 5 W's. To answer *who*, *where*, or *what* the mention refers to, the most important step of EL is not to identify an entity mention, but to correctly *link* it to its referent KB entry. The main challenge in EL is ambiguity: a single entity mention (e.g., “*Michael*”) can refer to multiple real-world entities in the KB (e.g., *Michael Jordan*, *Michael Jackson*, or even any other real-world entity that may not be described in the KB). At the same time, a single KB entity may

be referred to by multiple entity mentions (e.g., *Michael Jackson* may be referred to with “*the king of pop*,” “*MJ*,” “*Mr. Jackson*,” or simply “*Michael*”).

EL methods using Wikipedia have been effectively applied to many challenging domains where context is sparse, and language may be noisy, e.g., Twitter [176] and television subtitles [199]. However, the E-Discovery scenario provides an additional challenge over noisy and large volumes of data: the *entities of interest* may not be included in the reference KB.

In the first part of this thesis our *entities of interest* are these entities that are absent from the reference KB. Entities about which we may have little prior knowledge. We call these entities *emerging entities*, as they are initially absent from the KB, but of interest to an end-user, i.e., worthy of being incorporated into the KB.

We address three tasks that relate to emerging entities: The first is largely observational in nature, here, we study how entities emerge in online text streams, by analyzing the temporal patterns of their mentions before they are added to a KB. Next, we bootstrap the discovery of emerging entities in social media streams, by generating pseudo-ground truth to learn to predict which entity mentions represent emerging KB entities. And finally, we collect external and user-generated additional entity descriptions to construct and enrich entity representations to allow searchers to retrieve them more effectively.

Chapter 3. Analyzing Emerging Entities

We start our exploration of real-world entities in digital traces with a large-scale analysis of how entities emerge. The digital traces we study are online text streams (e.g., news articles and social media posts), and the entities of interest are emerging entities. Understanding the temporal patterns under which entities emerge in online text streams, before they are added to a KB may be helpful in predicting newly emerging entities of interest, or in distinguishing between different types of entities of interest. In Chapter 3 we study the temporal patterns of emerging entities in online text streams, and answer the following research question:

RQ1 Are there common temporal patterns in how entities of interest emerge in online text streams?

To answer this question, we analyze a large collection of entity time series, i.e., time series of entity mentions in the timespan between the first mention of an entity in online text streams and its subsequent incorporation into the KB. Our collection contains over 70,000 entities that emerge in a timespan of over 18 months. We find that entities emerge in broadly similar patterns. We distinguish between entities that emerge gradually after being introduced in public discourse, and entities that emerge more abruptly. Furthermore, we study the differences and similarities between how entities emerge in different types of online text streams, and the differences and similarities in the emergence of different types of emerging entities.

Chapter 4. Predicting Emerging Entities

Motivated by the findings of the dynamic nature of knowledge bases, and continuous stream of newly emerging entities in Chapter 3, in Chapter 4 we address the task of

predicting newly emerging entities. The digital traces we study are social media streams (i.e., Twitter), and our entities of interest are entities that are similar to a set of seed entities, i.e., entities that are described in a reference KB. In the discovery process a searcher may have some prior knowledge of the entities of interest, e.g., she may have a reference list of entities with the goal of exploring a document collection in search for similar entities. In such a scenario, where the entities of interest are absent from the KB, traditional EL methods fall short. In Chapter 4 we propose a method that leverages the prior knowledge that is encoded in a reference KB, and existing methods for recognizing and linking KB entities, to recognize newly emerging entities that are not (yet) part of the KB. In this chapter, we answer the following research question:

RQ2 Can we leverage prior knowledge of entities of interest to bootstrap the discovery of new entities of interest?

We propose an unsupervised method, where we use an EL method and an incomplete reference KB to generate pseudo-ground truth to train a named-entity recognizer to detect mentions of emerging entities. Mentions of emerging entities are entity mentions that were not linked by the EL method, but that do occur in similar contexts as those that are in the reference KB. We compare the effect of different sampling strategies on the pseudo-ground truth, and the resulting predictions of emerging entities. We show that sampling based on textual quality and the confidence score of the EL method are effective methods for increasing the effectiveness of discovering absent entities. Furthermore, we show that with a small amount of prior knowledge, our method is able to cope with missing labels and incomplete data, justifying the approach of generating pseudo-ground truth. Finally, the method we propose is domain and/or language independent, as it does not rely on language or domain-specific features, making it particularly suitable for the discovery scenario.

Chapter 5. Retrieving Emerging Entities

Finally, in Chapter 5, the last chapter of Part I, we address the task of *retrieving* entities of interest. As seen in Chapter 3, entities may suddenly appear in bursts, as events unfold in the real world. In order to capture the changing contexts in which entities appear, and improve the retrieval (i.e., search) of entities, we propose a method that dynamically enriches and expands the (textual) representations of entities. We enrich representations of an entity by collecting descriptions from a variety of different dynamic sources that represent the *collective intelligence*. The different sources constitute different (user-generated) digital traces, e.g., social media, social tags, and search engine logs. In expanding their representations with these collective digital sources, we bridge the gap between the informal ways how users search for and refer to entities, and the more formal ways of how they are represented in the KB. In this chapter, we answer the following research question:

RQ3 Can we leverage collective intelligence to construct entity representations for increased retrieval effectiveness of entities of interest?

We combine the collected additional descriptions from digital traces into a single entity representation, by learning to weight and combine the heterogeneous content from the

different sources. Our method learns directly from users' past interactions (i.e., search queries and clicks), and enables the retrieval system to continuously learn to optimize the entity representations towards how people search for and refer to the entities of interest. We find that incorporating dynamic description sources into entity representations enables searchers to better retrieve entities. In addition, we find that informing the ranker of the "expansion state" of the entity overcomes challenges related to heterogeneity in entity descriptions (i.e., popular entities may receive many descriptions, and less popular entities few), and further improves retrieval effectiveness.

Part II. Analyzing and Predicting Activity from Digital Traces

In the second part of this thesis we take a different view on entities of interest. Here, our entities of interest are the people who leave behind the digital traces, i.e., the producers of digital traces. Our aim is to better understand the producers' context under which they leave behind digital traces. This part revolves around finding evidence of activity in the real world, which is an essential task in the sense-making process of E-Discovery [197]. We present two case studies of analyzing and predicting human activity given digital traces.

Analyzing and predicting user behavior has a rich history in the information retrieval and the user modeling, adaptation and personalization communities [124]. Understanding human behavior by studying historic interactions and digital traces finds many applications, e.g., improving personalization and recommendation systems [157, 158], improving search engines [4, 8, 122], or improving information filtering [188].

In both case studies we look at the impact of the *contexts* in which digital traces are created, and their *content*, on predicting activity of entities of interest. In our first case study, we analyze and predict email communication in an enterprise. In our second case study, we study interaction logs of users with an intelligent assistant on a mobile device, and aim to predict user task execution.

Chapter 6. Analyzing and Predicting Email Communication

Analyzing communication patterns can be helpful in answering questions like *who was involved?*, and *who knew what?*, which are central questions in the discovery process. Increased understanding of the aspects that guide communication, and being able to predict communication flows can be helpful in identifying atypical or unexpected communications, a valuable signal in the discovery process [200]. In this chapter, the digital traces under study are enterprise email, and the entities of interest are emailers.

We study the impact of contextual and content aspects of email in predicting communication. More specifically, as context, we study the impact of leveraging the enterprise's "communication graph" for predicting email recipients. The communication graph provides contextual clues of the creation of digital traces, e.g., the "position" of an emailer in a communication graph may implicitly capture their position in the company, and the strength of ties, or the proximity between two emailers may capture their professional or social relations. Next, we study the impact the content of the digital traces, i.e., we leverage the similarity between emails for predicting likely recipients. In this chapter, we answer the following research question:

RQ4 Can we predict email communication through modeling email content and communication graph properties?

To answer this question, we present a hybrid model for email recipient prediction, that leverages both the information from the communication graph of the email network, and the Language Models (LM) of the emailers, estimated with the content of the emails that are sent by each user. We find that both the context and content of digital traces provide a strong baseline for predicting recipients, but are complementary, i.e., combining both signals achieves the highest accuracy for predicting the recipients of email. We obtain optimal performance when we incorporate the number of emails a recipient has received so far, and the number of emails a given sender sent to a recipient at that point in time in our model.

Chapter 7. Analyzing and Predicting Task Reminders

Our second case study revolves around a proliferating type of digital traces; user interaction logs with mobile devices, and more specifically, user interactions with a personal intelligent assistant. With the rise of mobile devices in E-Discovery, an ever important task is to understand real-life behavior of people through traces from their mobile devices.

Intelligent assistants are becoming ubiquitous; Google's Now, Apple's Siri, Amazon's Echo, Facebook's M, and Microsoft's Cortana have all been introduced in rapid succession over the last few years. In this chapter, we study user interaction logs with Microsoft's Cortana. Through their personal, human-like and conversational nature, intelligent assistants are more closely embedded in a user's daily life and activity than some of the technologies they borrow from (e.g., search engines, calendar, planning, and time management apps). By closing the gap between a user's offline and online world, intelligent assistants have a potentially large impact on a user's life, such as on her productivity, time management, or activity planning. The digital traces left behind through interacting with personal assistants may thus contain important signals related to the users' offline real-world activities. We focus on studying Cortana's reminder service, as reminders represent tangible traces of people's real-life (planned) activities and tasks. Understanding the temporal patterns related to reminder setting and task execution can help in inferring and understanding people's behavior in the real world.

We aim to identify common categories of tasks that people remind themselves of, and study temporal patterns linked to the types of tasks people execute. In this chapter, we answer the following research question:

RQ5 Can we identify patterns in the times at which people create reminders, and, via notification times, when the associated tasks are to be executed?

More specifically, we apply a data-driven analysis to identify a body of common tasks types that give rise to the reminders across a large number of users. We arrange these tasks into a taxonomy, and analyze their temporal patterns. Furthermore, we address a prediction task, and much like the work presented in Chapter 6, we study the impact of both the content and contexts of the digital traces. We show that the time at which a user creates a reminder (context) is a strong indication of when the task is scheduled to be executed. However, including the description of the task reminder (content) further improves prediction accuracy.

1.2 Main Contributions

In this section, we list the theoretical, technical, and empirical contributions made in this thesis.

Analyses and Algorithms

A large-scale study of emerging entities In Chapter 3 we study a large dataset which contains over 70,000 entities that emerge in a timespan of over 18 months in over 579M documents, and show that entities emerge in broadly two types of patterns: with an initial burst of increased attention leading up to incorporating in Wikipedia, or a more gradual pattern, where the attention builds up over time. Furthermore, we identify characteristic differences between how entities emerge in news and in social media streams. Finally, we show that specific entity types are more strongly associated with specific emergence patterns.

Unsupervised method for generating pseudo-ground truth using EL In Chapter 4 we propose an unsupervised method that uses an EL method for generating training material for a named-entity recognizer to detect entities that are likely to become incorporated in a KB. Our method can be applied with any trainable Named-Entity Recognition (NER) model and EL method that is able to output a confidence score for a linked entity. Furthermore, our method is not dependent on human annotations that are necessarily limited and domain and language specific, is not restricted to a particular class of entities, and is suitable for domain and/or language adaptation as it does not rely on language specific features or sources.

Dynamic collective entity representations In Chapter 5 we employ collective intelligence to construct “dynamic collective entity representations,” i.e., we dynamically create entity representations that encapsulate the different ways of how people refer to or talk about the entity. In doing so, we show we can bridge the gap between the words used in a KB entity description and the words used by people to refer to entities.

Generative hybrid model for email recipient prediction In Chapter 6 we propose a hybrid generative model aimed at calculating the probability of an email recipient, given the sender and the content of the email. We show how both the communication graph properties as email content properties contribute to a highly accurate prediction of email recipients.

Large-scale study of user interaction logs with a personal intelligent assistant In Chapter 7 we present the first large-scale study of the creation of common task reminders. We show that users largely remind themselves for short-term chores and tasks, such as to go somewhere, communicate, or perform daily chores. We develop a taxonomy of types of time-based reminders, facilitated by the data we have about the actual reminders created by a large populations of users. We show how reminders display different temporal patterns depending on the task type that they represent, the creation time of the reminder, and the terms in the task description.

We study temporal patterns in reminder setting and notification, demonstrating noteworthy patterns, which we leverage to build predictive models. The models predict the desired notification time of reminders, given the reminder text and creation time.

Empirical Contributions

A large-scale study of emerging entities In Chapter 7 we apply a hierarchical agglomerative clustering method to over 70,000 entity document-mention time series, and uncover two distinct patterns of emerging entities. We discover two distinct patterns of how entities emerge in public discourse.

Two sampling methods for pseudo-ground truth Chapter 4 presents two sampling methods for automatically generated training data. We show that (i) sampling based on the textual quality improves performance of NER and consequently performance of predicting emerging entities, and (ii) sampling based on the confidence score of the EL method which provides the pseudo-training data labels results in fewer labels but better performance.

Effect of KB size in emerging entity prediction In Chapter 4 we study the effect of the size of the KB (i.e., the amount of prior knowledge) on predicting newly emerging entities, and find consistent and stable precision regardless of KB size, which justifies our emerging entity prediction method that assumes incomplete data by design.

Dynamic collective entity representations In Chapter 5 we show entity representations enriched with external descriptions from various sources better capture how people search for entities than their original KB descriptions. Furthermore, we show that informing the ranker of the expansion state of the entity, i.e., the number and type of additional descriptions the entity holds, increases retrieval effectiveness.

Email recipient prediction In Chapter 6 we show that combining communication graph with email content features achieves optimal predictive power. We show that the number of email received by a recipient is an effective method for estimating the prior probability of observing a recipient, and the number of emails sent between two users is an effective way of estimating the “connectedness” between these two users, and is a helpful signal in ranking recipients.

1.3 Thesis Overview

The first chapter, which is the one you are currently reading, introduces and motivates the research topic of the thesis: automated methods for sense-making of digital traces. Furthermore, this chapter provides an overview of the main contributions, content, and origins of the further chapters in this thesis. Chapter 2 discusses the background and related work that serves as a base for the following chapters. We describe Information Retrieval (IR), Semantic Search, Entity Linking (EL), and user modeling and analysis. The core of this thesis consists of two parts.

In Part I of this thesis, we study emerging entities and incomplete knowledge bases. It consists of three chapters. Chapter 3 presents an empirical study of temporal patterns of entities as they emerge in online text streams, before they are incorporated into Wikipedia. Chapter 4 presents a method of leveraging an incomplete reference KB to generate pseudo-training data for training a system for discovering emerging entities, similar to the reference entities but absent from the KB. Chapter 5 presents a novel method of leveraging collective intelligence and learning from past behavior of users to construct dynamic entity representations, aimed at improving entity retrieval effectiveness.

Part II of this thesis revolves around analyzing and predicting human behavior from digital traces. Chapter 6 presents a case study of enterprise email communication, and provides insights into the different aspects that guide communication between people. Chapter 7 presents a large-scale user log study of an intelligent personal assistant, and shows the types of task people tend to remind themselves about.

Finally, Chapter 8 concludes this thesis, where we summarize the content and findings of this thesis, discuss the limitations of the presented work, and briefly reflect onto future work.

The two parts of this thesis (Part I and Part II) are self-contained and form independent parts. For the reader's convenience, both the background chapter (Chapter 2), and the conclusions (Chapter 8) follow this structure, i.e., the relevant background material, conclusions, and future work parts are organized separately for each part, so that both parts can be read independently.

1.4 Origins

This thesis is based on five publications [101–105]. In addition, it draws on ideas from four others [97–100]. This section lists for each chapter the publication it is based on, as well as the contributions of the author and co-authors.

Chapter 3 is based on *The Birth of Collective Memories: Analyzing Emerging Entities in Text Streams*, currently under submission at the Journal of the Association for Information Science and Technology (JASIST). This publication is written with Odijk and de Rijke. Graus wrote the clustering and analysis code. Odijk contributed to the data parsing pipeline (i.e., adding timestamps to entity annotations). Graus performed the experiments and the analysis. All authors contributed to the text.

Chapter 4 is based on *Generating Pseudo-ground Truth for Predicting New Concepts in Social Streams* which was published at ECIR 2014 by Graus, Tsagkias, Weerkamp, Buit-inck, and de Rijke. Graus implemented the general pipeline and algorithms, Tsagkias and Weerkamp contributed code to the sampling methods. Graus performed the experiments and analyses. All authors contributed to the text.

Chapter 5 is based on *Dynamic Collective Entity Representations for Entity Ranking* which was published at WSDM 2016 by Graus, Tsagkias, Weerkamp, Meij, and de Rijke. Graus implemented the general pipeline and algorithms, Tsagkias and Weerkamp

contributed code to the sampling methods. Graus performed the experiments and analyses. All authors contributed to the text.

Chapter 6 is based on *Recipient Recommendation in Enterprises Using Communication Graphs and Email Content* which was published at SIGIR 2014 by Graus, van Dijk, Tsagkias, Weerkamp, and de Rijke. Graus implemented the general pipeline and algorithms, Tsagkias and Weerkamp contributed code to the sampling methods. Graus performed the experiments and analyses. All authors contributed to the text.

Chapter 7 is based on *Analyzing and Predicting Task Reminders* which was published at UMAP 2016 by Graus, Bennett, White, and Horvitz. Graus performed the log-analysis, developed the task type taxonomy, implemented the algorithms for task reminder execution time prediction. Graus performed the experiments and analyses. All authors contributed to the text. This work was done while on an internship at Microsoft Research.

Finally, this thesis also indirectly builds on publications on entity linking for generating user profiles for personalized recommendation of events [98], a context-based entity linking method [97], entity linking of search engine queries [100], and tweets [93]. And publications around Semantic Search for E-Discovery [99, 251].

2

Background

“Nescire autem quid ante quam natus sis acciderit,
id est semper esse puerum.”
—Cicero, *De Oratore*, XXXIV

The work presented in this thesis sits at the crossing between the fields Information Retrieval (IR) and Natural Language Processing (NLP). This thesis deals with enabling and improving access to, and retrieval of information, and with improving and leveraging computer systems’ ability to process, understand, and extract information from natural language. In this chapter, we present all relevant background and related work that serves as a basis for understanding this thesis. We start at the very basics, with a high level explanation of Information Retrieval and the current state of affairs and challenges, in Section 2.1. Next, we move on to explain Semantic Search, Entity Linking (EL), and Knowledge Bases in Section 2.2. In Section 2.3 we discuss work related to the first part of this thesis, on analyzing, predicting, and retrieving emerging entities. Finally, we discuss the related work for the second part of this thesis on analyzing and predicting user behavior from digital traces, in Section 2.4.

2.1 Information Retrieval

IR is most prominent in our everyday lives in the search engines people consider their entry point to the internet. Sometimes literally: a large share of queries issued to search engines are so-called *navigational queries* — queries that represent the user’s intent of visiting a specific website [31]. To illustrate: the 10 most frequent queries issued to the Yahoo search engine include: `facebook`, `amazon`, and `yahoo` [110]. Modern-day search engines have no trouble addressing these kind of information needs, quickly serving us websites from indices of many billions of “documents.”

But in web search, the landscape is rapidly changing. At Google, the volume of mobile search has surpassed that of desktop search [95]. Around 20% of queries from mobile devices in the US are voice input [208], and over half of the teenagers in the USA use voice input to issue their search queries. Mobile web search changes the game. Voice queries are longer on average, and use richer language than text queries [110]. Displaying

2. Background

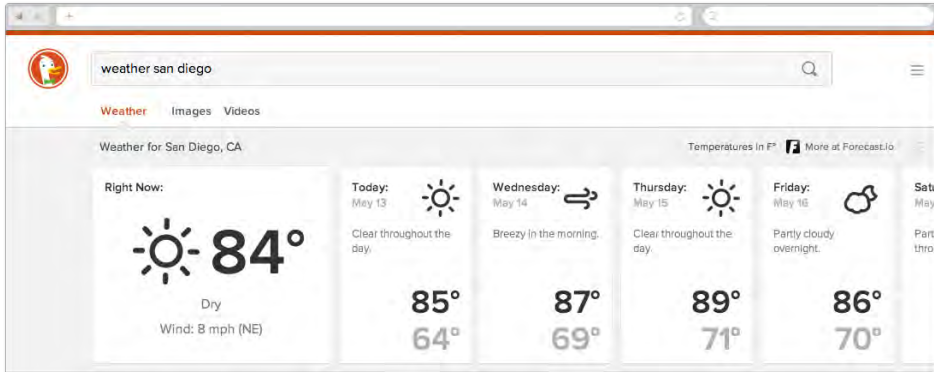


Figure 2.1: An example of a *knowledge card* displaying the weather forecast for San Diego on the search engine result page of DuckDuckGo.

results on a mobile device means a shift from the “10 blue link”-paradigm towards newer, richer, more reactive, and more interactive ways of presenting the user with results. One such example is the proliferation of *knowledge cards* in many current web search engines and intelligent personal assistants such as Apple’s Siri and Microsoft Cortana. Knowledge cards are small panels that summarize information to answer a query, e.g., the weather forecast, or central information around an entity of interest. See Figure 2.1 for an example knowledge card.

Even outside mobile web search, in the more “traditional” setting of desktop web search, natural language queries (and in particular natural language questions) are on the rise [262]. Compared to the aforementioned navigational queries, understanding the more complex information needs that come with longer queries is a more challenging task. Web search engines do not adequately support more complex exploratory search scenarios [115].

Outside the domain of web search, plenty more complex information needs and search tasks exist, e.g., (re-)finding email messages [37], desktop search [70], researching prior art in patents [132], or finding related work for a background chapter in a PhD thesis [249].

The shift towards more complex queries in web search and the proliferation of complex search scenarios in other domains are just two examples of where the traditional IR approach of matching words (in queries) to words (in documents), known as *lexical matching*, no longer suffices. More complex information needs call for a better understanding of these information needs; they call for moving beyond words, towards better understanding of the intents, needs, and meaning that underlies queries and search behavior. *Semantic search* [153] aims to fill this gap, by incorporating meaning (i.e., semantics), natural language structure, and (external) knowledge into the search process.

2.1.1 Retrieval Models

Before we delve into semantic search, we provide a brief overview of the traditional lexical matching-based retrieval models that we use in this thesis. Generally speaking,

a retrieval model aims to assign a score of “relevance” (itself a very poor-understood notion) to documents (d) from an index, given a user-issued query (q). The most naive way of quantifying how “relevant” a document is to a query, is to count the occurrences of the query words in the document. This query term frequency-based method suffers one drawback: not all words are equally important in a document [164]. Weighting terms according to their importance alleviates this.

TF-IDF

One such way of term weighting is to assign higher weights to terms that occur in fewer documents: the underlying assumption is that words that occur frequently in a document, but in few documents overall, are the most representative of the document. This assumption is modeled by combining a term’s frequency with its *inverse document frequency* [236]. This is the Term-Frequency, Inverse Document Frequency (*TF-IDF*) weighting scheme:

$$Score(q, d) = \sum_{t \in q} tf(t, d) \cdot \log \frac{N}{df_t}. \quad (2.1)$$

Here, q is the user issued query, d the document, $tf(t, d)$ the term frequency of query-term t in d , N the total number of documents (i.e., collection size), and df_t the document frequency of query-term t in the collection. This is a tried-and-tested, i.e., old, but still frequently used scoring method. Another scoring method (that represents the state-of-the-art in lexical matching-based retrieval), Okapi BM25 [222], shares the same notion of term frequency and inverse document frequency for estimating the relevance of a document to a query, with more elaborate techniques to account for query and document length.

Language Modeling and the Query-Likelihood Model

Another way to estimate the relevance of a document to a query is to use language modeling. A statistical language model (LM) is a probability distribution over words [224], which we can use to estimate the probability (i.e., how likely it is) that a query is generated by a particular LM (i.e., particular distribution). If we compute for each document in a collection its own LM, we can rank the documents based on the probability of generating query q . This ranking method is known as the *query likelihood model*, which postulates that documents with higher probabilities to generate the query should rank higher:

$$Score(q, d) = P(q | d) = \prod_{t \in q} P(t | d)^{n(t, q)}, \quad (2.2)$$

where the document LMs are derived from word occurrences in the documents, i.e., the probability of observing a word given a document ($P(w | d)$) is typically defined as:

$$P(w_i | d_j) = \frac{n(w_i, d_j)}{\sum_N n(w_i, d_j)}, \quad (2.3)$$

where $n(w_i, d_j)$ is the frequency of term w_i in document d_j . Using these simple word-counts yields a unigram LM, also known as the bag-of-words model, where the order

of words (in both the query and the document) does not affect the score. Extending this model to incorporate word order yields n -gram models.

While these models are simple and computationally cheap, they suffer one major drawback. Both TF-IDF and the query likelihood model, but also more advanced models, e.g., BM25, rely on lexical matching, i.e., words from the query need to appear in the documents to get non-zero relevance scores.

2.1.2 Evaluation

A fundamental aspect in IR is evaluation. In order to assess the performance and usefulness of novel retrieval models, methods, and algorithms, it is central to be able to compare the output of different methods (typically rankings), to measure differences and improvements. These evaluation metrics rely on the availability of *ground truth*, i.e., a sets of assessments (e.g., relevance scores for documents per query), usually collected beforehand, which can serve as a golden standard to compare against.

The topic of evaluation in IR is huge, and to address everything is beyond the scope of this thesis. Here, we restrict to listing and describing the evaluation metrics we employ in this thesis, which we break down into set-based, rank-based, and classification evaluation metrics. We point the interested reader to [165] for a more comprehensive overview.

Set-based Metrics

Precision corresponds to the fraction of retrieved items (e.g., documents) in the system's output that are relevant (i.e., the items that are in the *ground truth* set). It is computed by taking the fraction of True Positives (TP), i.e., the set of retrieved items that are in the *ground truth*, over all retrieved items of the system, i.e., the set of True Positives and False positives (FP), where False Positives correspond to the items that are in the system's output, but not in the *ground truth*, i.e., $\frac{TP}{TP+FP}$.

Recall is a related set-based metric, which corresponds to the fraction of relevant items in the system's output. It is computed by taking the fraction of True Positives (TP) over the set of True Positives and False Negatives (FN), where False Negatives correspond to items that are in the *ground truth*, but not in the system's output, i.e., $\frac{TP}{TP+FN}$.

We employ Precision and Recall to compute our method's effectiveness of predicting emerging entities in Chapter 4.

P@k is a variation of Precision that is helpful in ranking scenarios where the set of ground truth items is very large (e.g., in a web search scenario). P@k corresponds to the Precision of a system's output up to a certain rank (k). We employ P@k in Chapter 5.

Rank-based Metrics

Precision and Recall are set-based metrics, which do not take the ordering of results into account. To turn the set-based Precision into a metric that does take order into account, we can compute the **MAP** (Mean-average Precision). First, Average Precision (AP) is the average Precision over all ranks (i.e., all values of k), up to the point where recall is 1 (i.e., the rank at which all relevant items are retrieved). By subsequently averaging AP over multiple sets (e.g., over multiple queries), we yield MAP. We employ MAP in Chapter 5 and Chapter 6.

Classification Metrics

Accuracy is a classification metric, i.e., it measures how well a system assigns class labels to a set of samples, by computing the fraction of correct class labels (w.r.t. a set of ground truth labels) over all assigned class labels. We employ accuracy in Chapter 7, where we address a multiclass-classification task. We distinguish between micro- and macro-averaged accuracy. **Macro-averaged accuracy** does not take different distributions in class membership into account, and simply computes the accuracy by taking the total set of correct predictions over the total number of predictions. **Micro-averaged accuracy** does take different class distributions into account, by computing the accuracy per class, and averaging these class-specific accuracies over the number of classes.

2.2 Semantic Search

With the rise of more complex queries, information needs, search tasks, and challenging domains, the lexical matching-based retrieval models described above may not suffice. Semantic Search, or “search with meaning,” aims to move beyond the constraints of simple keyword matching, by incorporating additional “structure” in the search process. This can be in terms of semantic matching, e.g., by applying topic modeling methods to project documents and queries in some (lower dimensional) semantic space, where matching can happen without direct lexical overlap between query and document [153].

In this thesis, however, we focus on one particular form of additional “structure:” real-world knowledge. By incorporating knowledge from external sources, e.g., knowledge bases such as Wikipedia, one can enrich text and improve matching [52, 114]. But incorporating real-world knowledge also better supports the exploratory search process that is inherent to E-Discovery [26, 268]. In particular, the real-world entities that occur in documents are central to question answering from text (the questions being, e.g., the 5 W’s) [140].

2.2.1 Knowledge Bases and Entities

Given a KB that describes real-world entities and concepts (e.g., Wikipedia), EL addresses the task of identifying and disambiguating occurrences of entities in text. Or, more specifically:

Matching a textual entity mention, possibly identified by a named entity recognizer, to a knowledge base entry, such as a Wikipedia page that is a canonical entry for that entity [216].

EL is a key component in modern-day applications such as semantic search and advanced search interfaces, and plays a major role in accessing and populating the Web of Data [25]. It can also help to improve NLP tasks [91], or to “anchor” a piece of text in background knowledge; authors or readers may find entity links to supply useful pointers [116]. Another application can be found in search engines, where it is increasingly common to link queries to entities to present entity-specific overviews [9], or to improve ad-hoc retrieval [52, 114].

EL can be traced back to several related tasks that preceded it, such as record linkage [80] in the Database community, and coreference resolution [235] in the NLP community. However, in its current form EL has received a lot of attention from the IR community, as well as industry, since the Text Analysis Conference (TAC) Knowledge Base Population track introduced an EL task in 2009 [174].

The EL task can formally be described as follows: given an entity mention m (a term or phrase) occurring in reference document d , identify the entity e from a knowledge base KB that is the most likely referent of m . In this thesis, EL plays a central role in Part I (i.e., in Chapters 3, 4, and 5). EL consists of two distinct steps: (i) recognizing mentions of entities in the knowledge base in text (*entity mention detection*), and subsequently (ii) linking them to their referent knowledge base entries (*entity disambiguation*). We briefly explain the core challenges and methods for both steps.

Entity mention detection. The first step, known as entity mention detection, aims to identify the word sequences that refer to entities (*entity mentions*). One approach is to apply *named-entity recognition* (NER). NER has a rich history, and has been studied since the 1990s in the Message Understanding Conferences [107]. Years later NER gained more traction as part of the CoNLL shared tasks, where winning systems achieved around 90% accuracy of recognizing named-entities in news articles [243]. NER methods typically rely on learning patterns in sequences of words, by leveraging the structure of language, e.g., the functions of words and their surrounding words (entity mentions are typically proper names [243]), or features relating to their surface form (e.g., in many languages capitalized words are more likely to refer to entities).

An alternative approach for *entity mention detection* is so-called lexical matching, or dictionary-based mention detection [230]. This method leverages the rich metadata of a KB such as Wikipedia, by creating a lexicon or dictionary that maps entities (i.e., Wikipedia page IDs) to entity surface forms (i.e., entity mentions). These surface forms are extracted from the rich variety of different ways used to refer to Wikipedia pages. For example, a single entity on Wikipedia can be represented by its: title, anchor texts from hyperlinks on other Wikipedia pages (i.e., the text that is used to hyperlink from one Wikipedia page to another), and redirect pages (mostly manually-added links that cover (common) misspellings or alternative names of Wikipedia pages), e.g., the Wikipedia page for Kendrick Lamar has redirect pages such as K-Dot (Kendrick Lamar’s former stage name), and also misspellings, such as Kendrick Lamarr and Kendrick Llama.

After extracting all these candidate entity mentions and generating the mappings of surface forms to entities, statistics on their usage can be further leveraged to estimate the probability that a (sequence of) word(s) is a reference to a KB entity (i.e., an entity mention). One commonly used and intuitive method of estimating the probability that an n -gram is an entity mention is the so-called *keyphraseness*. It boils down to the fraction of the number of times an n -gram is used to refer to an entity (i.e., the n -gram is used as an anchor text in Wikipedia), over the number of times the n -gram occurs in Wikipedia (including the number of times it is not part of a hyperlink) [181]. To illustrate, the phrase *Kendrick* appears 5,037 times in Wikipedia articles,¹ of which it is part of a hyperlink

¹Wikipedia dump dated June 2014

(only) 24 times, resulting in *Kendrick's* keyphraseness score of $\frac{24}{5,037} = 0.005$. In contrast, the phrase *Kendrick Lamar* appears 698 times, of which 501 times as a hyperlink, yielding a keyphraseness score, i.e., the prior probability of the phrase *Kendrick Lamar* being an entity mention, of $\frac{501}{698} = 0.718$.

Lexical matching-based EL approaches have shown to be successful in the challenging domain of social streams [176], and have shown to be suitable for adaptation to new genres and languages [199]. As an added advantage, these methods are independent of language-dependent linguistic annotation pipelines, which are prone to cascading errors [85]. Their drawback, however, is that they are unable to identify entity mentions that are not in the lexicon, e.g., misspellings not covered by Wikipedia's redirect pages.

Entity disambiguation. Given the detection of (candidate) entity mentions, the next step is to assign the referent entities from the KB to the mentions, or alternatively, determine that the mention refers to an entity that is not in the KB. Also known as *entity disambiguation*, this task is commonly approached by one of the following two approaches: (i) *local* and (ii) *global* entity disambiguation methods.

Local methods attempt to resolve a mention (i.e., assign the correct entity to a mention) by only considering properties of a (candidate) entity mention and the candidate KB entities. A common local method of disambiguating entity mentions is by employing the *commonness*-score [181]. Commonness is a simple fraction of the number of times the entity mention is used as an *anchor* for a specific entity e , over the number of times it is used as an anchor to any entity in the KB [175]. Continuing with the previous example, the entity mention "*Kendrick*" is used as an anchor text of a hyperlink 24 times, of which in 8 cases, the hyperlink points to the Wikipedia page *Kendrick, Idaho*, i.e., the probability that the mention "*Kendrick*" refers to *Kendrick, Idaho* is $\frac{8}{24} = 0.333$, whereas the probability that the mention refers to *Kendrick_Lamar* is $\frac{2}{24} = 0.083$.

Commonness thus favors the most common target entity for a mention, which makes it work well in practice, particularly in domains that refer to popular entities by nature (e.g., news streams). Because of the emphasis on single occurrences of entity mentions, local EL methods have been shown to be highly effective in domains in which context is limited and/or noisy, e.g., microblog posts [176]. Additional methods have been proposed to expand context when it is limited [33, 42].

There is an increasing interest in approaches that link multiple entities in a document simultaneously [130]. These so-called *global* methods are based on a notion of "coherence" in the set of entities in a document. There are several methods for defining the "coherence" of a set of entities. One is to rely on the structural information of the KB, e.g., the link or category structure [182], leveraging categories and contexts [47], or by using graph-based metrics [109]. The TAC Knowledge Base Population (KBP) track has seen many systems that incorporate global approaches to EL [41, 48, 81, 130, 211]. Also, several publicly available EL frameworks that aim to resolve each entity in a document apply global methods, e.g., 'DBpedia Spotlight' [177] and the 'GLOW' framework [217]. The most common global approach uses entity "relatedness" [182]. Relatedness is based on the overlap of the sets of "related" entities of a pair of entities. Related entities can be, e.g., all entities that are linked to or from the Wikipedia page that represents the entity of interest.

A known challenge of global methods is that entity mentions in the document can themselves be ambiguous, i.e., the number of pairwise comparisons between candidate

entities for multiple mentions is exponential in their number. The problem of optimizing the global coherence function is NP-hard [47]. A common strategy to deal with this problem is to decrease the search space, e.g., by only considering non-ambiguous entity mentions (i.e., mentions that are only associated with a single target entity) [182]. While this evidently reduces complexity, the presence of such entities cannot always be assumed. Another approach is to resolve the ambiguous entity mentions to their most popular candidate [209]. Cucerzan [47] introduces a method for joint disambiguation that does not consider all combinations of assigning one disambiguation per surface form at a time. Instead, all possible disambiguations of a surface form (candidates) are considered simultaneously. Another method is to set a threshold on the number of entities to consider per mention (e.g., at most 20 candidate entities [217]). Ratnov et al. [217] train a local approach to resolve individual entity mentions, before applying the global approach for coherence.

Entity linking in social streams. The domain of social streams brings additional challenges to the EL task. Whereas typical NER approaches perform in the upper 90% range in terms of accuracy on “clean,” grammatically correct and correctly spelled news articles, the accuracy drops considerably when performing NER on misspelled, short, and noisy text, such as social media posts. The noisy character and unreliable capitalization of social streams degrades the effectiveness of NER methods [61, 62, 85]. One approach to addressing this is to tailor the linguistic annotation pipeline to Twitter, which relies heavily on large amounts of training data [27, 221]. Another approach to combat the lack of context, proposed by Cassidy et al. [42], is to expand a social media post with additional posts of the author, and additional posts found by a clustering algorithm, before optimizing the EL system’s output for global coherence. Finally, incorporating spatiotemporal features [78] has been shown to improve EL effectiveness in social streams. The method we follow in this thesis however, embraces the little context and ignores the notion of global coherence, opting instead for machine learning methods with features that focus on learning mappings between individual n -gram and entities [176].

2.3 Emerging Entities

In Part I of this thesis, we focus on analyzing, predicting, and retrieving “emerging entities,” i.e., entities that are not (yet) part of the KB. This falls into the domain of KB construction (KBC) or (cold start) knowledge base population (KBP), which encompasses the ambitious goal of creating a KB from raw input data (e.g., a collection of documents). KBP and KBC span many smaller subtasks, e.g., (new) entity mention detection, entity clustering, relation extraction, and slot filling [69, 193].

We narrow our focus, and address the analysis, prediction, and representation of emerging entities in three successive chapters. More specifically, in Chapter 3 we study whether there exist common temporal patterns in how entities emerge in online text streams, in Chapter 4 we propose a method that leverages EL to generate training data for predicting new mentions of emerging entities in social streams, and finally, in Chapter 5 we study a method for improving entity ranking by mining additional entity descriptions from different external sources, e.g., social media and search engine logs.

2.3.1 Analyzing Emerging Entities

In the first chapter, we analyze entities as they “emerge;” i.e., we study the time series of entity mentions in online text streams in the time span between their first mention (i.e., the entity surfaces) and the moment at which the entity is incorporated into the KB.

Previous work on studying the expansion of Wikipedia through the addition of new pages studies the phenomenon from the perspective of Wikipedia itself, e.g., by analyzing how newly created articles fit in Wikipedia’s semantic network, studying the relation between activity on talk pages and the addition of new content to articles, or by studying controversy and disagreement on new content through “edit wars” [134, 137, 272, 273].

Studying emerging entities has received considerable attention from the natural language processing and information retrieval communities. Most notably, different methods and systems for identifying and linking unknown or emerging entities have been proposed [120, 155, 192, 255]. More recently, Färber et al. [79] formalize and analyze the specific challenges and aspects that come with linking emerging entities, while Reinanda et al. [218] study the problem of identifying relevant documents for known and emerging as new information comes in.

2.3.2 Predicting Emerging Entities

Addressing the task of detecting entities that are not in the KB, i.e., identifying emerging entities, can be traced back to the introduction of the “NIL clustering” subtask in the TAC KBP Entity Linking track in 2011 [130]. The task was a straight-forward EL task: participants were given a reference KB (derived from Wikipedia), a collection of documents (mostly news articles), and offsets for entity mentions that occur in the documents. The task was to decide for each mention whether it refers to an entity that exists in the reference KB (and if yes, which), or whether it refers to a “NIL entity:” an entity that is not in the KB. In the latter case, the system needs to assign a new ID (NIL ID) to the mention, and assign other mentions that refer to the same underlying real-world entity to the same NIL ID (i.e., cluster mentions to NIL IDs). Since common EL methods rely on ranking candidate entities to mentions, the inclusion of mentions for out-of-KB entities is typically approached by learning a threshold (for the EL system’s confidence or retrieval score) to determine whether or not the mention refers to an entity already described in the KB. This approach was chosen, e.g., by [35, 144, 192].

In the TAC KBP Entity Linking track the entity mention detection step, shown to be the bottleneck in state-of-the-art EL systems [111], was not part of the task. However, in a realistic scenario, when entity mentions are not given, Wikipedia lexicons and/or lexical matching-based approaches cannot detect unseen entity mentions of new entities, which means different approaches need to be leveraged, e.g., named-entity recognition (NER). The method for detecting emerging entities with ambiguous names, presented by Hoffart et al. [120], leverages lexical matching methods for detecting entity mentions. However, in this approach, only ambiguously named emerging entities can be detected, making it complementary to the method we present in Chapter 4, which focuses on detecting new (i.e., unseen) mentions of emerging entities. Lin et al. [155] learn to identify entity mentions of absent entities by leveraging Google Books’ n-gram statistics. Their method relies on part-of-speech tagging to identify noun phrases in documents, and deciding for

each noun phrase that cannot be linked to a Wikipedia article whether or not it refers to an (absent) entity. Another way to model the task of detecting emerging entities is *entity set extraction*, where given a seed list of entity mentions, the task is to retrieve more entities that are similar to the seed set [203]. However, the task differs as it does not rely on a KB, and consequently, the rich structure and metadata that comes with it.

Automatically generated pseudo-ground truth. The approach we propose in Chapter 4 revolves around automatically generating training data using an EL system. Automatically generating training data for machine learning methods has been a longstanding and recurring theme in machine learning, dating back to the 1980s [241].

The idea of using EL systems for generating pseudo-ground truth, or *silver standard* training data, is not new either. Guo et al. [108] used an EL system to collect additional context words for entities, to aid the disambiguation step. Their mention detection step relies on the Wikipedia lexicon, meaning their system is unable to detect unseen entity mentions. Similarly, Zhang et al. [278] apply EL to improve the disambiguation step in EL, by replacing unambiguous mentions in entity linked documents with ambiguous ones to generate artificial but realistic training data. Zhou et al. [279] generate training data from Wikipedia to improve the disambiguation step as well. More specifically, they consider Wikipedia links as positive examples, and generate additional negative samples by taking each other entity that may be referred to by the same anchor text.

Attempts have been made to simulate human annotations for generating training data for named-entity recognition, too. Kozareva [143] uses regular expressions for generating training data for NERC. Nothman et al. [195] leverage the anchor text of links in between two same articles in different Wikipedia translations for training a multilingual NERC system. Wu et al. [269] investigate generating training material for NERC from one domain and testing on another. Becker et al. [21] study the effects of sampling automatically generated data for training NER.

What is new to the method for predicting mentions of emerging entities we propose in Chapter 4, is that it directly optimizes for the task at hand, i.e., we generate training data for a named-entity recognizer to detect a specific type of entity, i.e., entities that are worthy of being added to the KB.

2.3.3 Constructing Entity Representations for Effective Retrieval

A task closely related to EL is entity ranking: in the end, most EL systems approach the disambiguation problem as an entity ranking problem. However, there is a huge difference between ranking entities for a candidate entity mention, and ranking entities for a user-issued query. In particular, there may be a vocabulary gap between how users refer to entities, and how entities are described in knowledge bases. The task of retrieving the correct entity for user-issued queries is of central importance in the general web search domain [36], but also in a discovery or exploratory search setting, where analysts may be interested in retrieving entities with partial names, or nicknames.

As with many NLP and information extraction tasks, research into entity ranking too has flourished in recent years, largely driven by benchmarking campaigns, such as the INEX Entity Ranking track [57–59], and the TREC Entity track [13, 14]. Entity ranking is commonly addressed — much like EL — by exploiting content and structure

of knowledge bases, for example by including anchor texts and inter-entity links [230], category structure [12, 226], entity types [135], or internal link structure [252].

More recently, researchers have also started to focus on using “external” information for entity ranking, such as query logs, i.e., users’ past search behavior. Billerbeck et al. [24] use query logs of a web search engine to build click and session graphs, and walk this graph to answer entity queries. Mottin et al. [190] use query logs to train an entity ranking system using different sets of features. Hong et al. [123] follow a similar line of thought and enrich their knowledge base using linked web pages and queries from a query log.

The entity ranking task is usually approached in a static setting, in which web pages and queries are added to the knowledge base before any ranking is performed. One of the few initial attempts to bring in time in entity ranking is a position paper by Balog and Nørnvåg [11], who propose temporally-aware entity retrieval, in which temporal information from knowledge bases is required. Finally, closing the loop, EL can be applied to queries to improve entity ranking [114]. Entity ranking is inherently difficult due to the potential mismatch between the entity’s description in a knowledge base and the way people refer to the same entity when searching for it.

Document expansion. One way to bridge this gap is to expand (textual) entity representations with terms that people use to refer to entities. Collecting terms associated with documents, and adding them to the document (document expansion) is in itself not a novel idea. Previous work has shown that it improves retrieval effectiveness in a variety of retrieval tasks. Singhal and Pereira [232] are one of the first to use document expansion in a (speech) retrieval setting, motivated by the vocabulary mismatch introduced by errors made by automatic speech transcription. Ever since, using external sources for expanding document representations has been a popular approach to improve retrieval effectiveness.

In particular, it was shown that anchors can improve ad-hoc web search [73, 180, 259, 270]. Kemp and Ramamohanarao [138] state that document transformation using search history, i.e., adding queries that lead to the document to be clicked, brings documents closer to queries and hence improves retrieval effectiveness. Similarly, Xue et al. [271] study the use of click-through data by adding queries to clicked document representations. In this case, the click-through data includes a score that is derived from the number of clicks the query yields for a single document. Gao et al. [92] follow a similar approach, but add smoothing to click-through data to counter sparsity issues. Amitay et al. [6] study the effectiveness of query reformulations for document expansion by appending all queries in a reformulation session to the top- k returned documents for the last query. Scholer et al. [226] propose a method to either add additional terms from associated queries to documents or replace the original content with these associated queries, all with the goal of providing more accurate document representations.

Looking at other sources for expansion, Bao et al. [15] improve web search using social annotations (tags). They use the annotations both as additional content and as a popularity measure. Lee and Croft [147] explore the use of social anchors (i.e., content of social media posts linking to a document) to improve ad hoc search. Noll and Meinel [194] investigate a variety of “metadata” sources, including anchors, social annotations, and search queries. They show that social annotations are concise references to entities and outperform anchors in several retrieval tasks. Efron et al. [71] show that document

expansion can be beneficial when searching for very short documents (tweets).

As an alternative to expanding entity representations, more recently, advances in neural nets and deep learning have given rise to (continuous) entity representations, i.e., entity embeddings [28, 156, 234]. One of the drawback of these methods is that it is not trivial to employ them in an online, dynamic scenario, where representations may continually change.

Fielded retrieval. A common approach to incorporate different document expansions into a single document representation is to create fielded documents [223, 277]. Based on fielded documents, a variety of retrieval methods have been proposed. Robertson et al. [223] introduce BM25F, the fielded version of BM25, which linearly combines query term frequencies over different field types. Broder et al. [32] propose an extension to BM25F, taking into account term dependencies. Svore and Burges [239] use a machine learning approach for learning BM25 over multiple fields, the original document fields (e.g., title and body) and so-called “popularity” fields (e.g., anchors, query-clicks). Macdonald et al. [162] compare the linear combination of term frequencies before computing retrieval scores to directly using retrieval scores in the learning to rank setting and show that it is hard to determine a clear winner. Pérez-Agüera et al. [206] note how one of the challenges in structured IR is the fact that field importance differs among collections and that different collections means field importance needs to be optimized per collection.

2.4 Analyzing and Predicting Activity

In Part II of this thesis, we study the link between their digital traces and activities. Mining digital traces has a rich history in exploratory analyses and improving search engines and online products, as discussed in Chapter 1. More specifically, large-scale user logs from many users have been used for a range of purposes to improve online services and advance our understanding of how people use systems. Search engine queries and search-result clicks have been used to understand how people seek information online [260], train search engine ranking algorithms to better serve user needs [4, 131], and more generally, teach us about how humans behave in the world [220]. Although large-scale log analysis of online behavior has typically focused on search and browsing activity, recent work has targeted the large-scale usage of communication tools too, such as instant messaging [148] and email [142].

In this thesis, we focus on mining digital traces with the aim to understand the underlying behavior that gives rise to the traces. We address an email prediction task in Chapter 6 and aim to gain insights into usage patterns with the reminder service of an intelligent personal assistant in Chapter 7.

2.4.1 Mining Email

Email has a rich history as an object of study in computer science, once more spurred by the availability of data, specifically, the acquisition and subsequent release of the “Enron corpus,” by computer scientist Andrew McCallum [141] for a reported sum of \$10,000 [168].

Research initially focused on information extraction tasks such as recognizing speech acts [38, 44], people-related tasks, such as name recognition and entity resolution [66, 76, 183, 184], contact information extraction [10, 49], identity modeling and resolution [75], discovery of peoples' roles [150], and finally email-related tasks, such as automated classification of emails into folders [22], e.g., by leveraging the social network features of the email network [274]. The Enron email collection was also used very directly in an E-Discovery task in the TREC Legal [46, 117] tracks, which revolved around finding responsive documents for a given production request (in litigation).

Studying email collections as communication graphs (or social networks) has yielded insights into how communication patterns change and relate to real-world events (e.g., how interpersonal communication intensified during Enron's crisis period [67]), how important nodes (= email addresses) or influential members of the network can be discovered by looking at network properties [231], and how one can discover shared interests among members of an email network [228].

More recently, research interest in email has been rekindled. Spurred by publications from industry, where access to huge amounts of real email data and traffic has resulted in work on automated classification of email into folders [106], predicting recipients' follow-up actions for emails [51, 65], or predicting an email's (personalized) "importance" [2], and inferring the activities that govern email communication [210].

Finally, work on recipient prediction (the task we address in Chapter 6) generally included the additional constraint that one or more seed recipient is given, a task also known as CC-prediction [127, 202]. Previous attempts at the type of recipient prediction we address leverages either the communication network, constructed from previously sent emails [225], or the content of the current email [39]. Finally, motivated by privacy concerns previous work typically addresses recipient prediction by restricting to a sender's ego network for prediction (i.e., only considering the local network of the sender, as opposed to the global communication network). In Chapter 6 we ignore these constraints, motivated by our offline, batch, discovery scenario, and rather attempt to gain insights into enterprise-wide communication patterns, and in particular the importance of communication graph properties and email content features.

2.4.2 User Interaction Logs

As mentioned in Chapter 1, interaction logs with personal intelligent assistants may provide rich digital traces for inferring people's real life activities. In the case of intelligent assistants, analyzing user logs may support inferring the users' intents [7] or current activities [204]. Where previous work addressed modeling long-term goals [18], the Cortana reminder logs we study in Chapter 7 may help us in understanding short term goals. To study the relation between a user's reminders and real life activities and tasks, several areas of research are relevant. We focus largely on research on memory and (completing planned) tasks, and review research in the following areas: (i) reminders, (ii) memory aids, and (iii) prospective memory.

Reminders. A number of systems have been developed to help remind people of future actions [63, 64, 133, 146, 172], many of which leverage contextual signals for more accurate reminding. These systems can help generate reminders associated with a range

of future actions, including location, events, activities, people, and time. Two of the most commonly supported types of reminders are location- and time-based (and combinations thereof [63, 169]). Location-based reminders fire when people are at or near locations of interest [160, 245]. Time-based reminders are set and triggered based on time [83, 119], including those based on elapsed time-on-task [43]. While time-based reminders can provide value to many users, particular groups may especially benefit from time-based reminders. These include the elderly [173], those with memory impairments [136], and people seeking to comply with prescribed medications [113]. In Chapter 7 we study time-based reminders in the Cortana reminder service, and omit location and person-based reminders, which are less common in our data, and more challenging to study across users as they rely heavily on personal context and relationships between the user and the locations and persons that trigger the reminders.

Memory aids. Memory aids help people to remember past events and information. Studies have shown that people leverage both their own memories via recall strategies and the use of external memory aids to increase the likelihood of recall [129]. Aids can assume different forms, including paper [163] to electronic alternatives [17, 82, 219]. One example of a computer-based memory aid is the Remembrance Agent [219], which uses words typed into the text processor to retrieve similar documents. People also leverage standard computer facilities to support future reminding (e.g., positioning documents in noticeable places on the computer desktop) [17], which is inadequate for a number of reasons, including that the reminder is not pushed to the user [83]. Other work has focused on the use of machine learning to predict forgetting, and the need for reminding about events [133]. Cortana is an example of an interactive and intelligent external memory aid. Studying usage patterns and user behavior lets us better understand users' needs, developing improved methods for system-user interaction and collaboration, and more generally, enhance our understanding of the types of tasks where memory aids are necessary.

Prospective memory. Prospective memory (PM) refers to the ability to remember actions to be performed at a future time [30, 72]. Beyond simply remembering, successful prospective memory requires recall at the appropriate moment. PM failures have been an area of study [74, 229]. Studies have shown that failures can be linked to external factors such as interruptions [50, 198]. Prospective tasks are usually divided into time-based tasks and event-based tasks [72]. Time-based tasks are tasks targeted for execution at a specific future time, while event-based tasks are performed when a particular situation or event occurs, triggered by external cues, e.g., a person or a location [30]. Laboratory studies of PM have largely focused on retention and retrieval performance of event-based PM, as this is straightforward to operationalize in an experimental setting. Time-based PM is a largely overlooked type in PM studies [68], as this type of "self-generated" PM is difficult to model in a lab setting. The Cortana reminder logs we study in Chapter 7 represent a real-life collection of time-based PM instances. They provide insights in the type and nature of tasks users are likely to forget to execute.

2.5 What's next

The rest of this thesis is structured in two self-contained parts that can be read independently.

In Part I of this thesis, our *entities of interest* are emerging real-world entities, i.e., entities that are not described (yet) in publicly available KBs such as Wikipedia. First, we study how entities emerge in online text streams in Chapter 3. Next, we set out to predicting newly emerging entities in social streams in Chapter 4. Finally, in Chapter 5 we address the dynamic nature in which entities of interests may appear in online text streams, and leverage collective intelligence to improve the retrieval effectiveness of real-world entities by capturing the changing contexts in which they may appear.

Next, in Part II we switch our focus, and our *entities of interest* are the producers of the digital traces. Here, we present two case studies. First, we analyze and predict email communication by studying the communication network and email content in Chapter 6, and then we analyze interaction data of users with an intelligent assistant, to predict when a user will perform an activity in Chapter 7.

Finally, we summarize our findings, formulate implications and limitations of our work, and highlight some areas for potential future work in Chapter 8.

Part I

Analyzing, Predicting, and Retrieving Emerging Entities

3

Analyzing Emerging Entities

“The two most important days in your life are the day
you are born and the day you find out why.”

—Mark Twain

3.1 Introduction

In the first part of this thesis we focus on *emerging* real-world entities, i.e., entities that are not described (yet) in a knowledge base. In this first chapter, we take a closer look at how these entities emerge. This chapter serves as an introductory study into emerging entities, our object of study, which we study in subsequent chapters in the contexts of predicting them in social streams (Chapter 4), and enabling their effective retrieval (Chapter 5).

We study entities that emerge in online text streams, and are subsequently added to Wikipedia, the largest publicly available and collectively maintained KB. Wikipedia has been dubbed a global memory place [205], where collective memories are stored [112]. The moment when an entity is incorporated into Wikipedia, i.e., stored in collective memory, can thus be considered parallel to the collective’s *discovery* of an entity. Studying emerging entities, hence, provides insights into what the collective considers *entities of interest*.

We take a macro view and aim to understand the properties of and circumstances under which entities of interest emerge. We answer the following research question: Are there common temporal patterns in how entities of interest emerge in online text streams? (RQ1).

Studying online text streams and Wikipedia allows us to make observations and gain insights into the process of emerging entities at a large scale. Every day, new content is being added to Wikipedia, with the knowledge base receiving over 6 million monthly edits at its peak [238]. These edits may appear under different circumstances. Domain experts may find information missing on Wikipedia, and take up the task of contributing this new information. Alternatively, as events unfold in the real world, new, previously unknown, and unheard-of entities (people, organizations, products, etc.) emerge into public discourse. These new entities emerge online in news articles and social media

3. Analyzing Emerging Entities

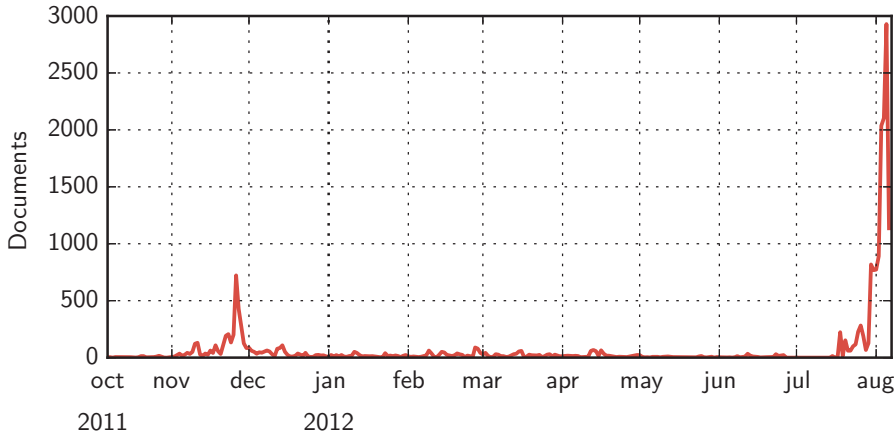


Figure 3.1: Emergence pattern of the entity `Curiosity (Rover)`, first mentioned in our text stream in October 2011. The Wikipedia page for `Curiosity` was created nine months later, on August 6, 2012. There are two distinct *bursts*, one late November 2011, the second shortly before the entity is added to Wikipedia. The two document bursts correspond to the Mars rover’s launch date (November 26, 2011) and its subsequent Mars landing (August 6, 2012). The time series shows us that while the launch did generate publicity, yielding a burst of documents (and signaling increased attention for the entity), at that point it was not deemed important enough to be added to Wikipedia.

postings that may describe or comment on events, e.g., the Olympics may introduce new athletes onto the world stage, a newly announced smartphone or video game console may generate a wave of activity on social media, or the opening of a new restaurant may be reported in local news media and pop up in social media.

In this chapter, we analyze a sample of online social media and news text streams, spanning over 18 months and comprising over 579 million documents. We focus on the *emergence patterns* of these entities, i.e., how a new entity’s exposure develops and evolves in the timespan between the entity’s first mention in online text streams, and when an article devoted to the entity is subsequently added to Wikipedia. More precisely, we define an entity’s *emergence pattern* to be its “document mention time series,” i.e., the time series that represents the number of documents that mention a specific entity over time, starting at the moment it is first mentioned, up to the moment it is incorporated into Wikipedia.

An example of one such emergence pattern is shown in Figure 3.1. The time series shows the number of documents that mention a given entity per day on the *y*-axis (i.e., the *emergence volume*), with the *x*-axis representing the time between the first mention of the entity in the online text stream and the day it is added to Wikipedia (i.e., the *emergence duration*).¹ Since emergence durations naturally differ between entities, our time series are of variable length. As we are interested in broad attention patterns, we study document mention time series, as opposed to total mention volume. And because we want to study

¹This is the English version of Wikipedia; see Section 3.3 for details on the data used in this chapter.

global, broad, and long-term patterns, our time series are at a granularity of days, not hours.

The main findings of this chapter are as follows. By clustering time series of mentions of entities as these entities emerge in news and social media streams, we find broadly two different emergence patterns: entities that show a *strong initial burst* around the time of their introduction into public discourse, and *late bursting* entities that exhibit a more gradual emergence pattern. Furthermore, we find meaningful differences between how entities emerge in social media and news streams: entities that emerge in social media streams exhibit slower emergences than those that emerge in news streams. Finally, we show how different types of real-world entities exhibit different emergence patterns; we find that the entities that emerge fastest are entity types that know shorter life-cycles such as devices (e.g., smartphones), and “cultural artifacts” (e.g., movies and music albums).

We proceed by formulating three sub research questions in Section 3.2. The data that we use in our study is described in Section 3.3. In Section 3.4 we detail the methods used to analyze the data. Results of our analysis are presented in Section 3.5. We conclude in Section 3.6.

3.2 Research Questions

In studying emergence patterns of entities, we apply three different methods for grouping entities, representing alternative views on emerging entities. In Section 3.5.1 below, we apply a burst-based unsupervised hierarchical clustering method to cluster similar entity emergence patterns, so as to discover groups of entities with broadly common emergence patterns. That is, we group entity time series by similarities in the peaks of comparatively higher activity (i.e., peaks of exposure in public discourse). This type of analysis is meant to help us answer the following research question:

RQ3.1 Can we discover distinct groups of differently emerging entities by clustering their emergence time series?

We show that entities emerge in two distinct patterns: so-called early bursting (EB) entities, that show a strong initial burst around the time of their introduction into public discourse, or late bursting (LB) entities, that exhibit a more gradual emergence pattern. EB entities are shown to exhibit shorter emergence durations than LB entities, i.e., they are deemed entities of interest more rapidly.

In Section 3.5.2, we adopt an alternative view of emerging entities, and examine their emergence patterns in different types of text stream, comparing patterns between entities that emerge in news to social media streams. This view is motivated by perceived differences in the nature of the professionally curated, authoritative, and high impact, “mainstream media” versus the user-generated, unedited, social media streams. We apply two grouping methods. First, we group the time series by type of text stream, and provide their descriptive statistics, hypothesizing that news streams will exhibit shorter emergence durations than social media streams, due to their reach and impact. We also analyze the cross-pollination between the two types of stream, i.e., we study whether entities appear first in either of the streams, or whether they appear in both at the same time, etc. These analyses help us answer the following research question:

RQ3.2 Do news and social media text streams exhibit different emergence patterns?

We find that news and social media streams show broadly similar emergence patterns for entities. However, entities that first emerge in social media streams on average are incorporated in the KB comparatively faster than entities that first emerge in news streams.

Finally, in Section 3.5.3, we study the similarities and differences between different types of entities as they emerge in public discourse. Specifically, we leverage DBpedia, the structured counterpart of Wikipedia, to group emerging entities by their underlying entity types, such as companies, athletes, and video games. This allows us to gain insights into whether different types of entities exhibit different emergence patterns. In addition, by considering entity types we study whether the news stream and the social stream exhibit different focal points, i.e., where professionally curated news streams exhibit a focus on the mainstream and user-generated social media streams surface more niche entities. This analysis helps us answer the following research question:

RQ3.3 Do different types of entities exhibit different emergence patterns?

We find that different entity types exhibit substantially different emergence patterns. Furthermore, we find that specific entity types are more or less likely to emerge in either the news or social media streams, which can be largely attributed to the different nature of both streams, and their authors. Finally, we find that entity types that have shorter emergence durations remain popular over time. The findings suggest that the entity types that are incorporated relatively fast in the KB, play a more central place in public discourse.

3.3 Data

As described in the introduction, we study the emergence patterns of entities by looking at the lead time between the first mention of an entity, i.e., its first appearance in online text streams, up to the moment they are incorporated into Wikipedia. To study these emergence patterns, we generate a custom dataset of timestamped documents that are annotated retrospectively with links to Wikipedia, including for each link (i) the creation date of the associated Wikipedia page, and (ii) whether the Wikipedia page existed at the time the document was created. Our dataset spans 7.3 million documents with 36.2 million references to 79,482 unique emerging entities. We create this dataset by extending an existing document-stream dataset (TREC KBA’s StreamCorpus 2014²) with an additional set of annotations to Freebase (FAKBA1³). We enrich the dataset by including Wikipedia creation dates of the Freebase entities, and the relative age of the entity to the document. We subsequently reduce this dataset by filtering out documents that have “future leaks,” i.e., that contain references to entities whose Wikipedia pages were created after the dataset’s timespan. Below, we describe our data acquisition and preparation processes in detail.

²<http://trec-kba.org/kba-stream-corpus-2014.shtml>

³<http://trec-kba.org/data/fakba1/>

3.3.1 Emerging Entities in News and Social Media

Our dataset is derived from the *TREC KBA StreamCorpus 2014*, a dataset of roughly 1.2 billion timestamped documents from various sources, e.g., blog and forum posts and newswires. It spans 572 days (or roughly 18 months) from October 7th, 2011 to May 1st, 2013 (line 1 in Table 3.1). The StreamCorpus is composed of multiple subsets, with slightly different periods and volumes (i.e., numbers of documents). All documents in the corpus that were automatically classified as being written in English have been automatically tagged for named entities with the Serif tagger [29], yielding roughly 580M tagged documents. This annotated subset of documents was the official collection for the TREC Knowledge Base Acceleration (KBA) task in 2014 [89].

Dalton et al. [53] further annotated these 580M documents with Freebase entities, resulting in the *Freebase Annotations of TREC KBA 2014 Stream Corpus, v1 (FAKBA1) dataset*, which spans over 394M documents (line 2 in Table 3.1). It is essential to our study that the Freebase dump used for linking the entities is dated after the span of the StreamCorpus. Because of this, we can isolate entities that are mentioned (i.e., linked) in documents before their corresponding Wikipedia page was created.

To extract Wikipedia page creation dates for the entities present in the TREC KBA StreamCorpus 2014, we first map the Freebase entities linked in the FAKBA1 collection to their corresponding Wikipedia pages, using the available mappings in Freebase. We extract Wikipedia page creation dates from a dump of Wikipedia with the full revision history of all pages (`enwiki-latest-page-meta-history.xml`). We append the Wikipedia page creation date or *entity timestamp* (denoted e_T) to each entity in FAKBA1. In addition, we include the entity’s “age” relative to the document timestamp (doc_T): the difference between the Wikipedia page creation date and the document timestamp, i.e., $e_{age} = e_T - doc_T$. The resulting dataset, FAKBA1, extended with the entity age and entity timestamp, is denoted *FAKBAT* (Freebase Annotations of TREC KBA 2014 Stream Corpus with Timestamps), see line 3 in Table 3.1.

As a next step, we filter to retain only documents that contain emerging entities. Emerging entities are entities with $e_{age} < 0$, i.e., entities mentioned in documents dated before the entity’s Wikipedia creation date. We denote the resulting subset of FAKBAT documents with emerging entities *OOKBAT* (Out of Knowledge Base Annotations (with) Timestamps). This yields a set of nearly 24M documents (line 4 in Table 3.1).

To be able to study an emerging entity’s complete emergence pattern, we take two additional filtering steps. First, we prune entities with creation dates later than the last document in our stream, to ensure the entities emerged in the timespan of our document stream, i.e., we remove all entities whose Wikipedia page has a creation date later than May 1st, 2013. Next, we prune all entities that are mentioned in fewer than 5 documents, to be able to visualize and study their time series. This yields our final dataset, which comprises 79,482 emerging entities (line 5 in Table 3.1).

Entity Types

In the analysis in Section 3.5.3, we leverage an entity’s “class” from the DBpedia ontology.⁴ Freebase provides mappings to Wikipedia and its structured counterpart DBpedia.

⁴<http://mappings.dbpedia.org/server/ontology/classes/>

3. Analyzing Emerging Entities

Table 3.1: Descriptive statistics of our dataset acquisition. Coverage over preceding dataset in brackets. Looking at the second and third row in the table, we note that roughly two-thirds of the FAKBA1 entities can be mapped to Wikipedia. However, this portion represents 98% of the mentions. Closer inspection showed that the missing one-third, i.e., the entities we could not map to Wikipedia creation dates, were Freebase entities that were not linked to their Wikipedia counterparts, most notably, WordNet concepts and entities from the “MusicBrainz” knowledge base (i.e., artists, albums, and artists). The last two rows show that one in ten of the entities emerge during the span of the dataset, however, they constitute a mere 1% of the mentions.

Dataset	# entities	# mentions	# documents
1. <i>TREC KBA</i> [89]	N/A	N/A	579,838,246
2. <i>FAKBA1</i> [53]	3,272,980	9,423,901,114	394,051,027 (68.0%)
3. FAKBAT	2,254,177 (68.9%)	9,221,204,641 (97.8%)	394,051,027 (100%)
4. OOKBAT	225,291 (10.0%)	94,929,292 (1.0%)	23,896,922 (6.1%)
5. Emerging entities	79,482 (35.3%)	36,242,096 (38.2%)	7,291,700 (30.5%)

In the DBpedia ontology, an entity is assigned to one or more classes in a tree-like class structure. We map each of our emerging entities to the classes assigned in DBpedia, e.g., the entity *Barack_Obama* is mapped to the *Person*, *Politician*, *Author*, *Award Winner* classes. We extract these mappings by extracting all triples that have `rdf:type` property (e.g., *Barack_Obama* `<rdf:type>` *Person*). Out of the 79,482 emerging entities in our dataset, we have 39,713 class-mappings (i.e., a coverage of 50.0%).

Entity Popularity

Finally, as a proxy for an entity’s popularity, which we use to study the composition of clusters and the different substreams in Section 3.5.3, we extract Wikipedia pageview statistics. More specifically, we extract the total number of pageviews each entity received during 2015. We choose to use the pageview counts of a year that falls outside of the timespan of our dataset so as to minimize the effects of timeliness (i.e., we want to separate the true “head” entities from the ones that have a shorter lifespan).

3.3.2 Time Series of Emerging Entity Mentions

We use the Emerging Entities dataset that we have created (line 5 in Table 3.1) to generate time series that describe their emergence. Figure 3.2 shows the document volume over time of the different streams in our dataset. It shows a larger number of news documents compared to social media documents in the first half of the data, and a larger number of social media compared to news documents in the second half. In total, the news stream comprises 1,836,022 documents, making it substantially smaller than the social media stream, at 5,357,014 documents. The news and social media streams themselves are composed of multiple datasets from different sources [88], which explains the gap seen around May 2012 up to somewhere in July 2012.

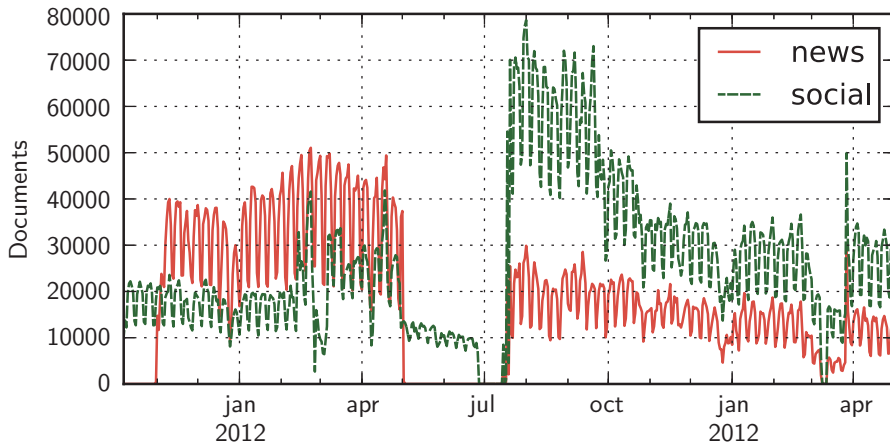


Figure 3.2: Total document volume (i.e., count) for the news (green line) and social media (grey line) text streams over time ($n = 23,896,922$). The streams are composed of multiple separate datasets, explaining the gap in May through mid-July of the news stream. (Best viewed in color.)

We note that 79,482 entities emerge during the 18+ month (572 days) timespan of the TREC KBA StreamCorpus 2014 dataset, i.e., on average over 138 new entities emerge per day. Looking at the distribution at which these entities emerge over time, we observe in Figure 3.3 that they do not emerge uniformly. In particular, Figure 3.3 shows a gradual increase of emerging entities between the start of our dataset and May 2012, at which it peaks. The subsequent gap can be explained by the absence of the news stream during that time (see also Figure 3.2).

The core unit in our analysis are so-called *entity document mention time series*, i.e., time series that represent the number of documents that mention an entity over time (see, e.g., Mars Curiosity’s document mention time series in Figure 3.1). These time series are characterized by several properties. First, the time series are of variable length: each entity’s time series starts at the first mention of the entity in our dataset, and ends at the day the Wikipedia page for that entity was created. For some entities, the time series may span several days, whereas others may span months. Second, the time series in our dataset are not temporally aligned. They exhibit different absolute timings, where the date of the first mention (i.e., the start of the time series) and last mention (i.e., the end of the time series) varies between emerging entities.

3.4 Methods

Our first research question, Can we discover distinct groups of differently emerging entities by clustering their emergence time series? (RQ1), revolves around discovering common emergence patterns. We apply a time series clustering method for discovering groups of entities with similar emergence patterns. In this section, we first describe

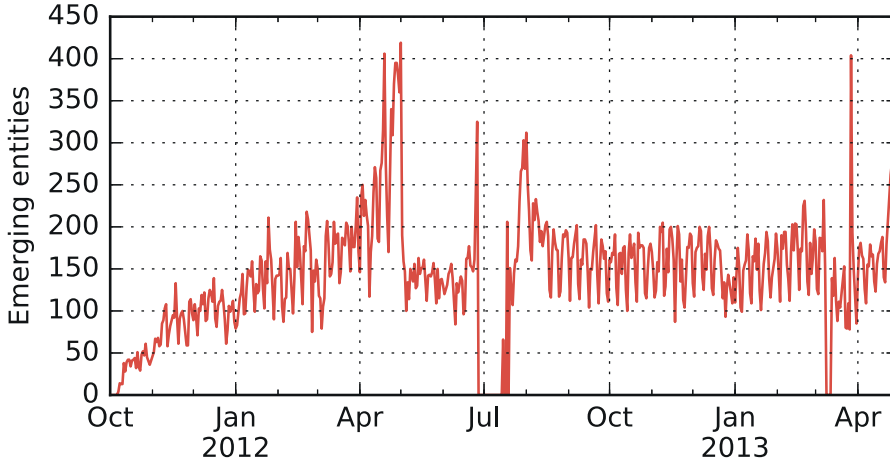


Figure 3.3: Distribution of emerging entities in our dataset ($n = 79,482$). The x -axis shows the timespan of our dataset in days, the y -axis shows the number of entities that emerge in our dataset on that day (i.e., have their Wikipedia pages created).

our time series clustering method, explain and motivate the choices for representing the emerging entity's time series. After that, we describe the general time series analysis methods that we use.

3.4.1 Time Series Clustering

Clustering time series consists of three steps: First, we need to normalize our time series as they might span very different periods of time. Next, we measure the similarity between time series by applying a similarity metric. And third, we apply a hierarchical agglomerative clustering method to group entities in groups with similar emergence patterns.

Normalization

As described in Section 3.3.2, the time series in our dataset might span different periods and are not temporarily aligned. For these reasons, we cannot rely on time series analysis and modeling methods that leverage aligned time series or seasonal patterns. Because of variable lengths, we cannot leverage similarity methods that assume a correspondence between the data points between two time series such as, e.g., Euclidean distance. Furthermore, to be able to visualize clusters and groups of similar time series, we linearly interpolate the time series to have equal length [214]. Finally, as we are interested in the similarity in emergence patterns, not in individual differences between popular and long-tail entities (i.e., absolute number of mentions), we standardize all time series by subtracting the mean and dividing by standard deviation to account for the differences in volume/popularity [253].

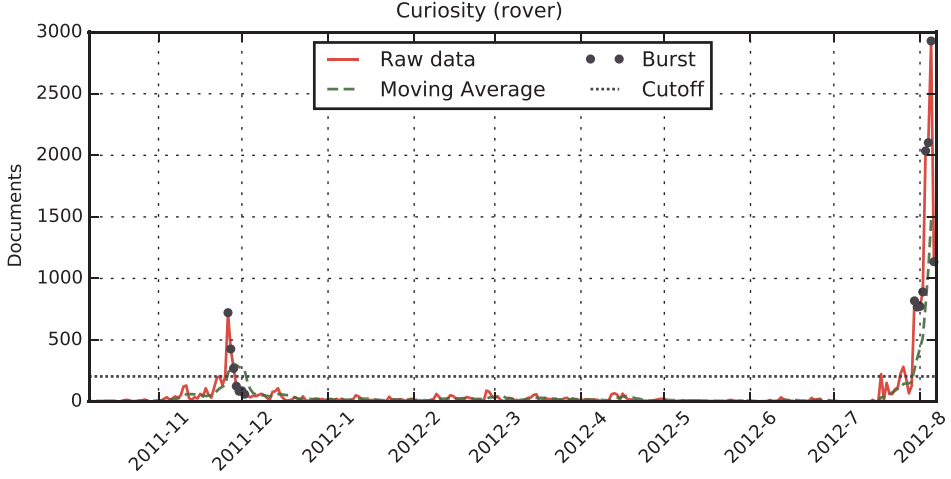


Figure 3.4: Emergence pattern of *Curiosity (rover)*. (Best viewed in color.)

Burst Similarity

Typically, time series similarity metrics rely on fixed-length time series, and leverage seasonal or repetitive patterns [154]. But as noted above, our time series are of variable length, and not temporally aligned. For this reason, common time series similarity metrics such as Dynamic Time Warping (DTW) are not applicable [23]. We are interested in the moments at which the attention or focus around an entity in public discourse increases, i.e., we are looking for periods with higher activity. These so-called time series “bursts” may be correlated to real-world activity and events around the entity.

To address the nature of our time series as well as our focus on bursts, we turn to BSim [253] (Burst Similarity) as the similarity metric we leverage to compare time series. It relies on detecting bursts, and using the overlap in bursts between time series as the notion of similarity.

For burst detection, we compute a moving average for each (raw) entity document mention time series (T_e), denoted T_e^{MA} . We set our parameter w , indicating the size of the rolling window to 7 days. Bursts are the points in T_e^{MA} that surpass a cutoff value (c). We set $c = 1.5 \cdot \sigma_{MA}$, where σ_{MA} is the standard deviation of MA . These parameter choices for w and c are in line with previous work [253]. Figure 3.4 shows an example time series (T_e), with the bursts detected for the emerging entity *Curiosity (rover)*. The detected bursts correspond to the earlier described launch and landing of the Mars Rover.

Hierarchical Agglomerative Clustering

Now that we can measure similarity between time series, we need to identify clusters of similar time series. To this end, we compute the Similarity Matrix (SM) with all pair-wise burst similarities. More specifically, to cluster emerging entities, we first apply L_2 normalization to SM , and then convert it to a distance matrix DM ($DM =$

1 – SM). Next, we apply hierarchical agglomerative clustering (HAC) on DM using the `fastcluster` package [191] to discover groups of similar time series at different levels of granularity. We apply *Ward’s method* as our linkage criterion. Ward’s method is an iterative approach, where one starts with singleton clusters, and aims to merge the pair of samples that maximally decreases the within-cluster variance at each successive iteration [257].

3.4.2 Time Series Analysis

Throughout this chapter, we take an exploratory approach to analyzing, visualizing, and comparing patterns of groups of time series to discover meaningful clusterings [254]. We apply different grouping strategies, e.g., implicit groups, such as the clusters we find by using our clustering method, or explicit groups, e.g., entities that emerge either in social media or in news streams.

We apply two analysis methods to study different groups of time series: (i) visualization of group signatures that represent common emergence patterns within a group of time series, and (ii) analysis of descriptive statistics that reflect properties of the underlying time series.

Visualization

To compare patterns and trends across different groups of emerging entities, we visualize and compare the so-called *group signatures*, i.e., the average of all time series that belong to a group. See Figure 3.6 below for an example of the group signature of all emerging entities in our dataset ($n = 79,482$).

Two challenges specific to the time series that we study in this chapter arise when visualizing them. First, their *duration*. The time series we study in this chapter (may) differ in length, as emergence durations differ between entities. Second, their *alignment*. The time series are not temporally aligned, as their start (i.e., $x = 0$) is marked when the entity is first mentioned in the online text stream, and end at the time at which the entity is incorporated into Wikipedia. For these reasons, we linearly interpolate the time series to the (overall) highest emergence duration, effectively “stretching” them to have equal length. Next, we align them in relative duration, i.e., the first and last mentions for each entity is set at the start and end of the x -axis, respectively. This allows us to visualize both the clusters themselves, and the corresponding cluster signatures.

Descriptive Statistics

While studying group signatures of time series allows us to discover similar patterns and study and compare broad patterns and trends, they do not paint the full picture. More fine-grained properties of emergence patterns, e.g., the average *emergence duration* (the time between a new entity’s first mention in the text stream and its subsequent incorporation into Wikipedia) and the *emergence volume* (the total number of documents that mention the entity as it emerges) are difficult to convey through visualization alone. In order to study these aspects, we represent the groups of time series through different descriptive features that reflect the emergence and burst behavior of the time series that belong to

Table 3.2: Descriptive statistics used for analyzing and comparing different groupings of emerging entities. We distinguish between *time series statistics* (top three) and *burst statistics* (bottom three). All statistics are computed for the period ranging from the emerging entity’s first mention in the corpus to the creation date of the Wikipedia page devoted to it. The burst durations and values are computed over the normalized time series (see Section 3.4.1).

Emergence volume	Number of documents that mention the entity
Emergence duration	Number of days from first mention to incorporation
Emergence velocity	$\frac{Volume}{Duration}$ (average number of documents per day)
Burst number	Total number of bursts
Burst duration	Normalized average durations of bursts (i.e., burst widths)
Burst value	Normalized average heights of bursts (i.e., burst heights)

a group. For an overview of the emergence and burst statistics that we consider, see Table 3.2.

Chi-square Goodness of Fit Test

In Section 3.5.3 we compare the distribution over entity classes per online text stream (i.e., comparing the entity classes in the social media stream to those in the news stream). To assess whether the differences in these class-distributions are statistically significantly different, we apply a chi-square goodness of fit test to both distributions. In addition, we rank the classes by their contribution to the difference, using *chi-grams*, i.e., we compute for each class: $\frac{observed - expected}{\sqrt{expected}}$, where *observed* corresponds to the number of entities that belong to a particular class in one set (e.g., the entities that emerge in social media) and *expected* corresponds to the number of entities that belong to a particular class in the global population (i.e., all entities that emerge in our dataset).

3.5 Results

In this section, we present the analyses that answer our three research questions. First, we study the time series clusters that result from our clustering method in Section 3.5.1. Next, in Section 3.5.2, we study similarities and differences between emergence patterns in social media and newswire streams. In Section 3.5.3, we study the underlying entity types, and their emergence patterns.

3.5.1 Emergence Patterns

The first research question we aim to answer, Can we discover distinct groups of differently emerging entities by clustering their emergence time series? (RQ1), is at the core of our study into emerging entities. Finding similar patterns and studying how an entity appears in online text streams before it is added to Wikipedia allows us to gain insights into the circumstances in which entities emerge, i.e., in which they are deemed *entities of interest*.

3. Analyzing Emerging Entities

Table 3.3: Global time series and burst descriptive statistics.

duration (# days)			volume (# docs)			velocity (docs/day)		
mean \pm	std	med.	mean \pm	std	med.	mean \pm	std	med.
245 \pm	153	221	183 \pm	1,180	32	0.87 \pm	5.6	0.19
n_bursts			bursts durations			bursts values		
3.8 \pm	2.62	3	0.03 \pm	0.03	0.02	0.03 \pm	0.08	0.02

Figure 3.5 shows a cluster tree that results from applying hierarchical clustering on the BSim similarity matrix computed for the time series of all entities under consideration, as explained in Section 3.4.1. The tree shows a clear subdivision into smaller clusters. At its highest level, the tree shows two distinct clusters, each of which is broken down in multiple smaller sub-clusters. The cluster tree in Figure 3.5 helps us to organize our analysis; in the following section, we look at different granularities of similarly emerging entities, by exploring three levels of this cluster tree: first, we study the *global emergence patterns*, by taking the time series that are at the root node of the tree (i.e., the first level of the tree, denoted *Top level* in Figure 3.5). Then we study the main two clusters, at the next level of the tree (denoted *Level 1* in Figure 3.5). Finally, move down another level in the tree and study the set of six clusters at *Level 2* in Figure 3.5.

Global Emergence Pattern

First, we examine the emergence pattern of the global average, by taking the group at the root node of the cluster tree (i.e., all time series in our dataset, where $n = 79,482$). Figure 3.6 shows the global emergence signature, and Table 3.3 shows the associated descriptive statistics.

Figure 3.6 shows how both the emerging entities’ introduction into public discourse (the first mention at the left-most side of the plot) and subsequent incorporation into Wikipedia (the right-most side of the plot) occur in bursts of documents, i.e., overall, the largest number of documents that mention a newly emerging entity are either at the start or at the end of their time series. This can be explained as follows. The entrance into public discourse represents the first emergence of an entity, whereas being added to Wikipedia is likewise likely to happen in a period of increased attention, e.g., a real-world event that puts the entity in public discourse. Between these two bursts, the number of documents that mention the entity seems to increase gradually as time progresses, suggesting that on average, the number of documents that mention a new entity, and thus the attention the entity receives in public discourse increases over time before it reaches “critical mass,” i.e., when the entity is deemed important enough to be incorporated into the KB.

Next, we look at the global emergence pattern’s descriptive statistics; see Table 3.3. Here, we see how on average it takes 245 days between an emerging entity’s first mention and being added to the KB. In these 245 days, an emerging entity is mentioned in 183 documents on average. However, both the emergence durations and emergence volumes show large standard deviations (of 153 days and 1,180 documents, respectively), indicating that they differ substantially between entities, further motivating our clustering approach

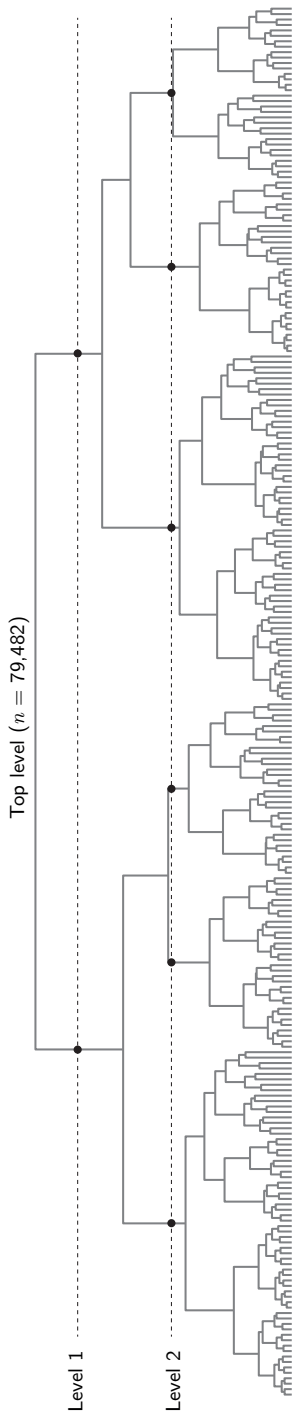


Figure 3.5: Dendrogram resulting from applying hierarchical agglomerative clustering using BSim [253] similarity, on our corpus of emerging entity time series ($n = 79,482$). The three cutoff-points at which we analyze the clusters are denoted Top level, Level 1 (2 clusters), and Level 2 (6 clusters). For clarity, the tree is truncated by showing no more than 7 levels of the hierarchy.

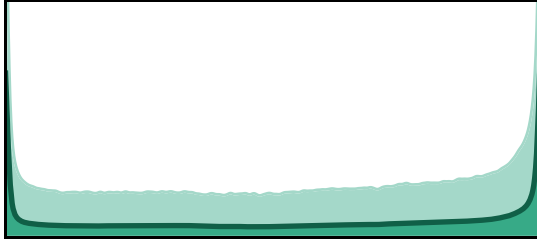


Figure 3.6: Global cluster signature (of all emerging entities) where $n = 79,482$. That is, the top level in the dendrogram in Figure 3.5. The axes are not labeled since all time series values (i.e., document counts) are standardized, and the series are linearly interpolated to have equal length. The solid line represents the cluster signature (i.e., the average time series), the lighter band represents standard deviation.

to zoom into the different underlying patterns. The emergence velocity shows how, on average, an entity is mentioned in less than a document a day.

To better understand how the documents that mention an emerging entity are distributed over time, e.g., to gain insights into whether they are concentrated in a few bursts, or spread out more uniformly over the timeline, we turn to the burst statistics in Table 3.3. On average, an entity is associated with 3.8 bursts, indicating that entities are likely to resurface multiple times in public discourse before being deemed important enough to be added to the KB. The average burst durations span a mere 3% of an emerging entity’s time series, indicating that emerging entities spend the majority of their time “under the radar.” The heights of these bursts (i.e., burst values) show a comparatively large standard deviation, suggesting that the heights differ substantially between different entities and bursts.

In summary, globally, entities experience a long time span between surfacing in online text streams and being incorporated into the KB; they are associated with multiple bursts and thus display a resurfacing behavior. Finally, the large standard deviations seen at the descriptive statistics suggests the entities show large variations in terms of their emergence patterns. Below, we study whether grouping the time series of new entities by similar burst patterns allows us to find groups of broadly similarly emerging entities in terms of their group signatures, and emergence and burst-features.

Clusters at Level 1 in Figure 3.5: Early vs. Late Bursts

In our first attempt at uncovering distinct patterns in which entities emerge, we look at the first two main clusters that appear at Level 1 of the cluster tree in Figure 3.5. The resulting cluster signatures are shown in Figure 3.7.

Much like the global cluster signatures we studied in the previous section, the clusters at Level 1 show two main bursts: the *initial burst* around the first mention, and the *final burst* around the last mention before an entity is added to Wikipedia. However, as we will show, the left cluster, which we call *early bursting (EB) entities*, is characterized

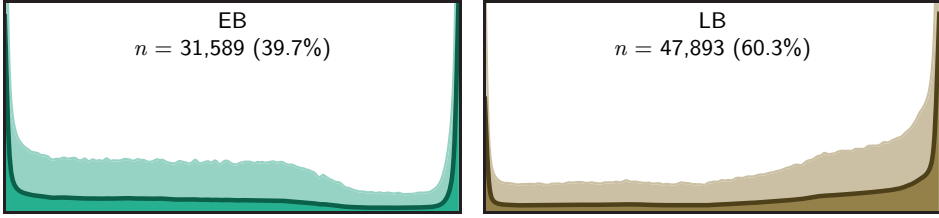


Figure 3.7: Cluster signatures of the early bursting entities (left plot) and late bursting entities (right plot) clusters, denoted Level 1 in Figure 3.5. The solid lines represent the cluster signature (average), the lighter bands represent the standard deviation.

by a stronger initial burst, with the majority of the documents that mention the entity concentrated at the time around the first mention when the entity surfaces in the online text streams. This suggests that the cluster contains new entities that suddenly emerge and experience a (brief) period of lessened attention, before being incorporated into the KB. The right cluster, which we denote as *late bursting (LB) entities*, shows a more gradual pattern in activity towards the point at which the entity is added to the KB, much like we saw in the global signature (Figure 3.6).

Looking at Figure 3.7, we note two main differences between the group signatures of the early bursting and late bursting entities. First, the way documents are distributed between the initial and final burst. The EB entities cluster show a more “abrupt” final burst: the signature shows the majority of the documents in the wake of the initial burst, i.e., at the left-hand side of the plot. In the next phase, the volume of documents seems to gradually wind down into a relatively quiet period, which finally seems to abruptly transition into the final burst at the right hand-side of the plot. In contrast, the LB entities cluster shows a relatively subtle initial burst, which like the EB-cluster appears to quiet down, followed by a gradual increase of document volume that leads up to the final burst.

A second difference is the height difference between the initial and final bursts. The EB cluster shows roughly equally high initial and final bursts. The LB cluster shows a substantially smaller initial burst, which suggests the introduction into public discourse is comparatively subtler than its addition, i.e., these entities may emerge “silently,” suggesting the entities are less central, more niche, and less widely supported. In summary, the first split separates the entities that are associated with a strong “initial” emergence in online text streams from the entities that more gradually build up to the moment at which they are incorporated into the KB.

Next, to better understand the different characteristics of the two clusters, we study the descriptive time series and burst statistics of the time series in the early bursting (EB) and late bursting (LB) entity clusters in Table 3.4. Before we proceed, we determine whether the differences between the statistics of the two clusters are statistically significant. To do so, we perform a Kruskal-Wallis one-way analysis of variance test, also known as a non-parametric ANOVA. Following this omnibus test, we perform a post-hoc test using Dunn’s multiple comparison test (with p-values corrected for family-wise errors using Holm-Bonferroni correction). Comparing all descriptive statistics from the two clusters, we find that all differences are statistically significant at the $\alpha = 0.05$ level.

3. Analyzing Emerging Entities

Table 3.4: Comparison of early bursting and late bursting entities clusters statistics.

	proportion	duration (#days)			volume (#docs)			velocity (docs/day)		
		mean \pm std	med.		mean \pm std	med.		mean \pm std	med.	
EB	0.40	224 \pm 146	195		118 \pm 804	22		0.70 \pm 6.45	0.15	
LB	0.60	259 \pm 156	238		225 \pm 1,371	42		0.99 \pm 4.96	0.23	
		n.bursts		burst durations			burst values			
EB	0.40	3.32 \pm 2.20		3	0.03 \pm 0.03	0.02	0.05 \pm 0.11			
LB	0.60	4.12 \pm 2.82		4	0.03 \pm 0.03	0.02	0.03 \pm 0.05			

Zooming into the statistics, we make the following observations. First, there are more LB than EB entities, at 60.3% vs. 39.7%. LB entities emerge more slowly than EB entities: the LB entities exist for a longer period in public discourse (with an *emergence duration* of 259 days, versus 224 for EB entities), and receive more attention (with an *emergence volume* of 225 documents, versus 118 documents for EB entities) before being deemed important enough to be incorporated into the KB. The EB entities’ shorter emergence durations and lower emergence volumes suggest a higher “urgency” or timeliness, suggesting this cluster may contain entities that represent sudden events, e.g., large-scale natural catastrophes or societal events, that will typically be incorporated in Wikipedia soon after they first emerge in public discourse. The descriptive statistics of the LB entities too may indicate they comprise less timely or urgent entities, e.g., recurring events, such as sports events that may appear in public discourse long before being part of the KB.

This view of the slower, less timely LB entities, and the more urgent, fast, and timely EB entities is supported by the burst statistics. First, the average burst heights of EB entities are higher (at 0.05 on average, versus 0.03 for LB entities), suggesting LB entities see a more evenly spread volume of documents that mention them. Next, EB entities show a lower number of bursts (3.22 on average, versus 4.12 for LB entities). Fewer and higher bursts, together with shorter emergence durations and lower emergence volumes supports the view of more urgent or timely EB entities, i.e., those that are more quickly (in terms of time and number of documents) incorporated into the KB, while exhibiting more bursty patterns. In the next section, we show that entities that emerge in news likewise exhibit higher burst heights and fewer bursts on average, further exploring the notion of higher urgency or importance.

In summary, we find that the two clusters at Level 1 in Figure 3.5 differ substantially and significantly in terms of their emergence patterns and burst properties. Our visual inspection of the cluster signatures and the analysis of burst and emergence features suggests LB entities emerge more slowly, i.e., build up attention more slowly before being added to the KB, whereas EB entities are associated with more sudden and higher bursts of activity, prior to being incorporated into the KB.

Clusters at Level 2 in Figure 3.5

Next, we look at the clusters at Level 2 in Figure 3.5, i.e., the different clusters that make up the LB and EB clusters. For brevity, we refer to these clusters as *subclusters*. Figure 3.8 shows the signatures of the six subclusters. In general, the distinction between faster EB entities, and comparatively slower LB entities, remains in the subclusters.

Using the procedure described in the previous section, we test whether the differences between the subclusters are statistically significant. Overall, the properties of time series within each subcluster differ significantly, except for early bursting 1 and late bursting 2a, which may not differ statistically significantly in terms of emergence volume. And in terms of burst statistics, some clusters show inconclusive differences. Specifically, early bursting 1 and 2a do not differ statistically significantly in terms of burst values, and early bursting 2a and late bursting 2b do not differ statistically significantly in terms of burst durations.

The cluster signatures in Figure 3.8 show that the entities that belong to the LB entities subclusters (the top three plots) exhibit an increase in document volume in the period leading up to the final burst before entering into the KB. This suggests that even when the entity first appears in online text streams with a relatively big burst, its subsequent incorporation into Wikipedia is not instantaneous. Consider, e.g., the previously shown example of the Mars Curiosity (in Figure 3.4); whereas its first burst happens around October 2011, it is added to Wikipedia after a relatively quiet period, 9 months later. Whereas the LB entities subclusters both differ in the “steepness” of the final burst and the relative moment of the increase of activity, the three EB entities subclusters (the bottom three plots) differ mainly in the moment of the increased activity, i.e., the “bump” in the plot.

One outlier exists in the LB entities cluster: late bursting 2a (second plot from the top). Its cluster signature shows a high and sudden burst of activity just before the entity is incorporated into Wikipedia, seemingly omitting the period of gradually increasing attention between the initial and final bursts seen at the other two subclusters. Its emergence features in Table 3.5 show that with an average duration of 146 days before an entity is added to the KB, it is almost twice as fast as the global average. Both the cluster signature and the average number of bursts (1.6 on average) show that the entities exhibit a single, steep, high volume burst before being incorporated into the KB. Compared to the other two subclusters in the same LB group, the number of bursts is substantially lower, and the average burst values are substantially higher. This suggests that the entities in the LB 2a cluster comprise abruptly or suddenly emerging entities, like events, and indeed, upon closer inspection, events such as the World Music Festival Chicago, the 2013 British Grand Prix, the 2012 Volvo World Match Play Championship, and the 2012 Sundance Film Festival belong to this cluster.

With the exception of this outlier, the entities in the LB subclusters show longer emergence durations and substantially higher emergence volumes on average, further supporting our distinction of slowly emerging entities. In particular, late bursting 1 and late bursting 2b (the third and first plot from the top, respectively), with 242 and 227 documents on average, with relatively long emergence durations (255 and 294 days on average, respectively), suggest that these clusters contain slowly emerging niche entities, that are not as widely supported by “the masses” as, e.g., the entities that are part of the LB 2a cluster. Manual inspection of the subclusters confirms this, revealing entities such as Summerland Secondary School, but also a long-tail of person entities (e.g., AFL athlete Rodney Filer, jazz musician Kjetil Møster) in LB 1, whereas entities that represent more “substantive” events are in LB 2a, e.g., the 2012 Benghazi attack, the Greek withdrawal from the Eurozone, and the Incarceration of Daniel Chong.

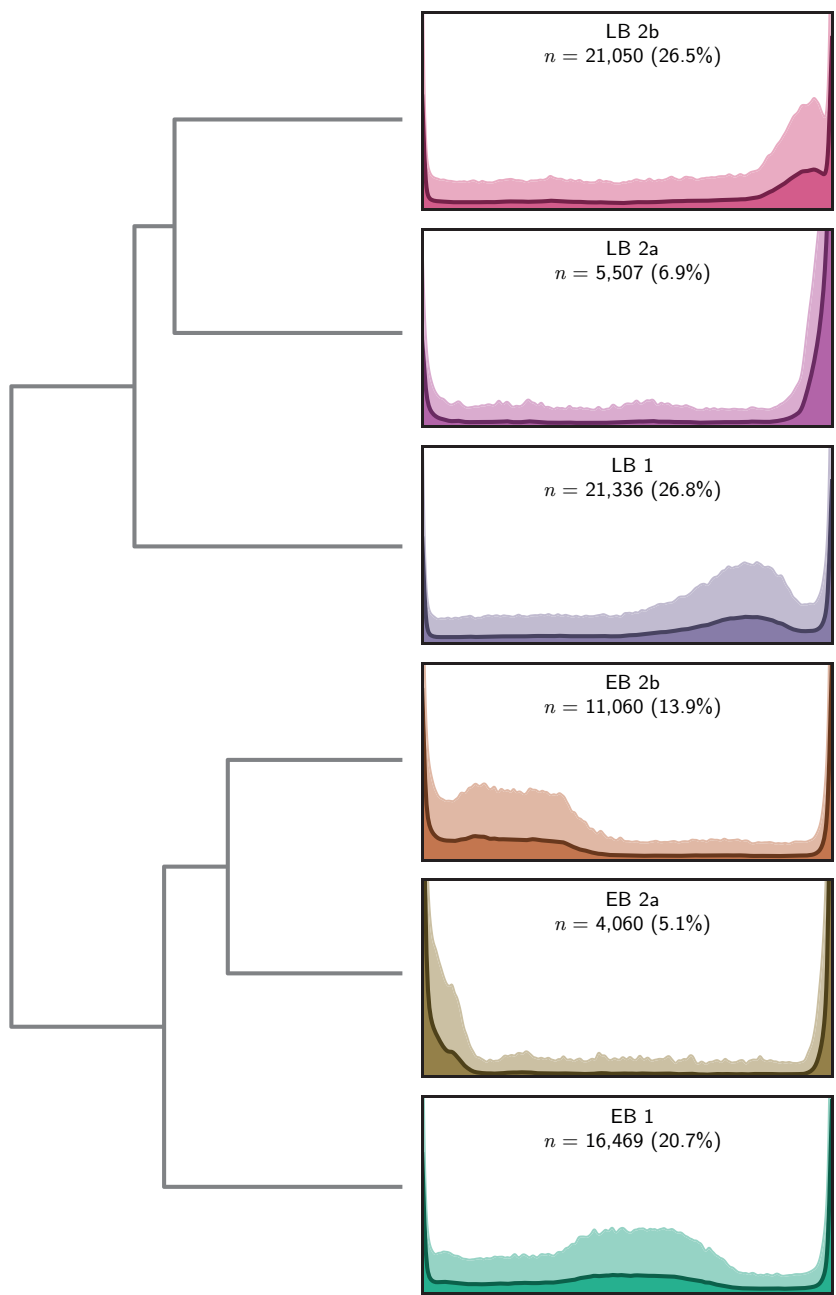


Figure 3.8: Signatures of the clusters at Level 2 in Figure 3.5. Solid lines represent the cluster signatures (average of all samples in the cluster), lighter bands represent standard deviation.

Table 3.5: Comparison of the entity cluster statistics for clusters at Level 2 in Figure 3.5.

	proportion	duration (#days)			volume (#docs)			velocity (docs/day)		
		mean \pm std	med.		mean \pm std	med.		mean \pm std	med.	
EB 1	0.21	238 \pm 148	223		145 \pm 868	28		0.83 \pm 8.53	0.17	
EB 2a	0.05	220 \pm 116	191		74 \pm 1,270	13		0.31 \pm 3.63	0.08	
EB 2b	0.14	204 \pm 150	164		94 \pm 363	19		0.65 \pm 2.37	0.16	
LB 1	0.27	255 \pm 148	230		242 \pm 1,660	43		1.01 \pm 4.94	0.22	
LB 2a	0.07	146 \pm 121	110		155 \pm 557	29		1.07 \pm 3.12	0.33	
LB 2b	0.26	294 \pm 158	297		227 \pm 1,183	45		0.95 \pm 5.36	0.20	
		n.bursts			burst durations			burst values		
EB 1	0.21	3.65 \pm 2.21	3		0.03 \pm 0.03	0.02		0.04 \pm 0.07	0.02	
EB 2a	0.05	2.26 \pm 1.27	2		0.02 \pm 0.02	0.02		0.10 \pm 0.22	0.02	
EB 2b	0.14	3.22 \pm 2.32	3		0.03 \pm 0.03	0.02		0.06 \pm 0.11	0.03	
LB 1	0.27	4.18 \pm 2.62	4		0.03 \pm 0.03	0.02		0.02 \pm 0.05	0.01	
LB 2a	0.07	1.60 \pm 0.88	1		0.04 \pm 0.03	0.04		0.05 \pm 0.05	0.05	
LB 2b	0.26	4.72 \pm 2.98	4		0.02 \pm 0.02	0.02		0.03 \pm 0.06	0.01	

Summary

In this section, we answer our first research question: “Can we discover distinct groups of differently emerging entities by clustering their emergence time series?”

We performed hierarchical clustering using a burst similarity-metric of the emerging entity time series. We discovered two distinct emergence patterns: early bursting entities and late bursting entities. Both the visual inspection of the cluster signatures, as the analysis of the descriptive statistics of the time series in the clusters support the same findings: the EB entities are characterized by fewer but higher bursts, with shorter emergence durations and lower emergence volumes. The LB entities on the other hand, seem to emerge more slowly, with a more gradual increase of exposure in the online text streams, before reaching the point at which the entity is added to the KB. This can be seen both in the cluster signature, as in the descriptive statistics, which show longer durations, higher volumes, and a larger number of bursts.

3.5.2 Entity Emergence Patterns in Social Media and News

In this section, we answer our second research question: “Do news and social media text streams exhibit different emergence patterns?” The news or social media document streams represent content from different sources: the news stream consists of traditional online news sources, where the content is mostly written by professional journalists; the social media stream contains mostly user-generated content, and consists of, e.g., forums and blog posts, but also content that was shared through other social media platforms such as Twitter and Facebook (through the bit.ly URL-shortener service). Studying whether different sources surface different entities, and exhibit different emergence patterns, can provide insights into how the nature of the source may affect the discovery process.

In the previous section, we have shown that 79,482 entities emerge in the combined

3. Analyzing Emerging Entities

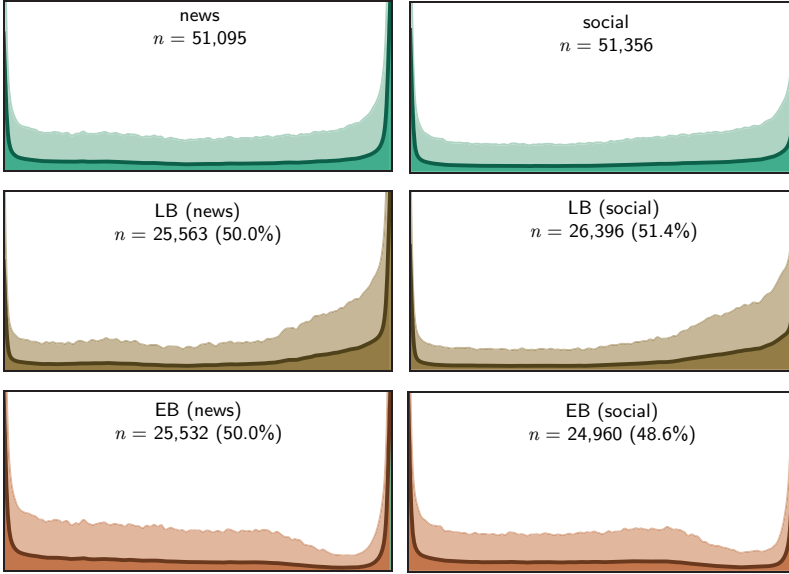


Figure 3.9: News vs. Social stream cluster signatures. The top row shows the global cluster signature of the news (left) and social (right) streams. The bottom two rows show the signatures of the late bursting and early bursting entity clusters for each stream (news left, social right).

news and social media streams. Taking a closer look by splitting out these entities by stream, we find 51,095 of these entities emerge in the news stream (i.e., are mentioned in the news stream), similar to the number of entities that emerge in the social media stream, at 51,356. Finally, 30,148 of the emerging entities are mentioned in both streams before being incorporated into Wikipedia.

In order to answer our second research question, we consider (i) the emergence patterns of entities that emerge in social media and news streams, and (ii) entities that appear in both streams vs. (iii) entities that appear in only one of the two streams.

Global: News vs. Social

First, we compare the emergence patterns of entities in news and social streams. We apply the same hierarchical clustering method as we have done in Section 3.5.1 on the two subsets of entities that emerge in news and social media streams (where $n_{news} = 51,095$ and $n_{social} = 51,356$).

Unsurprisingly, the emergence patterns are largely the same in the two streams and highly similar to the global patterns studied in Section 3.5.1. Both the news and social streams exhibit the same general global emergence pattern, witnessed by the largely similar clusterings we yield. Both streams exhibit groups that are similar to the early bursting and late bursting entities discovered in the previous section (shown in Figure 3.7).

Table 3.6: Global time series and bursts statistics per type of text stream.

stream	duration (#days)			volume (#docs)			velocity (docs/day)		
	mean \pm std	med.		mean \pm std	med.		mean \pm std	med.	
news	216 \pm 147	170		123 \pm 579	28		0.75 \pm 4.28	0.19	
social	234 \pm 153	211		65 \pm 239	19		0.48 \pm 3.33	0.12	
	n.bursts			burst durations			burst values		
news	3.14 \pm 2.10	3		0.03 \pm 0.03	0.02		0.04 \pm 0.07	0.02	
social	3.75 \pm 2.54	3		0.03 \pm 0.03	0.02		0.03 \pm 0.07	0.02	

Figure 3.9 first shows the global emergence patterns (top row, in green), and then the LB and EB entity clusters in the news (left) and social media (right) streams in the bottom two rows. Looking more closely at the global signatures of the news and social media streams, we note that entities that emerge in the news stream have slightly more of their emergence volume mass after the initial burst (i.e., the left-hand side of the plot), compared to the global pattern of the social media stream, which exhibits more of a gradual increase of emergence volume towards the final burst (i.e., the right-hand side of the plot). This may be attributed to the slightly higher proportion of early bursting entities in the news stream, which has 50.0% of its entities falling in this cluster, while the social media stream has 48.6%. The two main clusters (late bursting and early bursting) show broadly similar distributions and patterns, with the LB news cluster showing slightly more mass after the initial burst. This may be attributed to the “blind spot” of the news stream in our dataset (shown in Figure 3.2).

Table 3.6 shows the emergence and burst descriptive statistics of the streams. Statistical testing (using the same Kruskal-Wallis test followed by Dunn’s multiple comparison test with Holm-Bonferroni adjusted p -values used in Section 3.5.1) shows that the differences in the descriptive statistics between both streams are statistically significant at the $\alpha \leq 0.05$ level. With 216 days on average, entities emerge in news streams more quickly than entities in the social media stream. These shorter emergence durations are seen with higher emergence volumes on average: an entity that emerges in news is mentioned on average in 123 documents between their initial and final mention, nearly double the number of documents in social media (65). Recall that the total number of documents in the social media stream is larger, at 5.3M documents (versus 1.8M documents for the news stream — see also Figure 3.2). The higher number of documents with comparatively shorter emergence durations further supports the observation that emerging entities in news are picked up quicker than those that emerge in social media.

Furthermore, in the previous section, we have seen how early bursting entities exhibit fewer but higher bursts, and reasoned they represent more “urgent” or timely entities. The emergence features of entities emerging in news supports this notion of timeliness or urgency: they exhibit higher and fewer bursts on average.

In summary, we have shown that entities emerging in news and social media show broadly similar patterns, with both the cluster signatures and descriptive statistics being similar to the emergence patterns in the combined text streams in Section 3.5.1. We have also shown that news streams seem to surface entities more quickly than social

media streams, which we attribute to the different nature of the streams (professional and authoritative versus unedited and user-generated). In Section 3.5.3 we revisit this hypothesis by studying how the types of emerging entities differ between streams.

Who's First?

Of the 79,482 entities that emerge in the 18 month period our dataset spans, 30,148 appear in both the news and social media stream; 20,947 entities are mentioned exclusively in the news stream, never appearing in social media (*news-only*) between surfacing in online text streams and being incorporated into the KB. Finally, 21,208 appear only in the social media stream (*social-only*). See also Table 3.7.

Of the 30,148 entities that emerge in both streams, the majority appears in the social media stream before they appear in the news stream. This may be explained by the nature of the publishing cycles of the two streams; whereas traditional newswire has a more thorough publishing cycle, where stories need to be checked and edited before being published, social media — in particular forums and blog posts — follows a more unedited and direct publishing cycle.

The entities that appear in a social media stream first, which we denote *social-first*, cover 62.9% ($n = 18,967$) of the entities that emerge in both streams. The opposite pattern, where entities appear in news before they appear in social media (*news-first*), comprises 29.1% of the entities that emerge in both streams ($n = 8,794$). Entities that emerge in news first, subsequently appear in social media streams faster than vice versa: it takes a *news-first* entity on average 66 days to appear in the social media stream after surfacing in the news stream, whereas the other way around takes 49 days. The remaining comparatively small number of entities are mentioned in both streams on the same day (*same-time*): 7.9% ($n = 2,387$). The latter group of entities are expected to emerge more quickly, by virtue of appearing more widely in public discourse, and hence being more urgent and central.

The burst and emergence descriptive statistics, as recorded in Table 3.7, support this view: the *same-time* entities show comparatively short emergence durations. With an average 197 days between being first mentioned and being added to Wikipedia, *same-time* entities emerge substantially quicker than entities that appear in both streams at different times (at 298 days and 281 days on average for *news-first* and *social-first* entities respectively). Furthermore, the *same-time* entities exhibit higher emergence volumes too, accounting for the highest overall velocity at 2.87 documents per day, further supporting the hypothesis that these entities are more urgent and central.

Entities that emerge only in one of the two media streams show shorter emergence durations than those that appear in both but at different times (at 250 and 214 for news and social media respectively). These observations can be explained by the fact that longer duration means that it is more likely for an entity to cross from one stream to the other. Similarly, the longer average emergence durations of the *news-first* and *social-first* entities are paired with a larger number of bursts (around 4 on average, versus 3 on average for entities that emerge in either the news or social media stream). Much like what we saw in the previous section between late bursting and early bursting entities, the shorter durations and smaller number of bursts suggests that entities that

Table 3.7: Time series and burst statistics for entities that emerge in either or both streams (first). Burst and emergence features for our five groups of entities: entities that emerge in both streams, but are first mentioned in the news stream (*news-first*), entities that emerge in both streams, but are first mentioned in the social media stream (*social-first*), entities that emerge in both streams, and appear in both on the same day (*same-time*), entities that emerge only in the news stream (*news-only*), and finally, entities that emerge only in the social media stream (*social-only*).

stream	duration (#days)			volume (#docs)			velocity (docs/day)		
	mean \pm	std	med.	mean \pm	std	med.	mean \pm	std	med.
news first	298 \pm	139	305	123 \pm	291	53	0.58 \pm	1.59	0.21
social first	281 \pm	157	276	182 \pm	445	74	0.95 \pm	3.22	0.32
same time	197 \pm	147	163	192 \pm	662	67	2.87 \pm	23.59	0.51
only news	250 \pm	152	216	415 \pm	2,215	65	1.45 \pm	6.45	0.35
only social	214 \pm	148	190	33 \pm	134	12	0.41 \pm	2.60	0.08
	n.bursts			burst durations			burst values		
news first	3.92 \pm 2.50		3	0.02 \pm 0.02		0.02	0.04 \pm 0.08		0.02
social first	4.10 \pm 2.86		3	0.03 \pm 0.03		0.02	0.04 \pm 0.08		0.02
same time	4.25 \pm 2.86		4	0.02 \pm 0.02		0.02	0.04 \pm 0.08		0.01
only news	3.40 \pm 2.32		3	0.03 \pm 0.03		0.02	0.03 \pm 0.08		0.01
only social	2.99 \pm 2.08		2	0.03 \pm 0.03		0.02	0.04 \pm 0.08		0.02

emerge in a single stream are more central or important in public discourse.

Summary

News and social media streams show broadly similar emergence patterns for entities. However, while the patterns may be similar, the population and the behavior of entities emerging in news and social differ significantly. More specifically, entities emerging in the social media stream seem to do so slower on average than in news. Looking in more detail at the interactions between the two streams, we notice that entities that appear in both streams on the same day are the fastest to be incorporated into the KB. Furthermore, we find that entities that first emerge in social media are more quickly picked up in news streams than vice versa.

3.5.3 Emergence Patterns of Different Entity Types

In this section, we answer our third research question: “Do different types of entities exhibit different emergence patterns?” First, we analyze the descriptive statistics of each entity type in our dataset, to assert whether different types of entities exhibit different behavior in terms of how they emerge in online text streams. Next, we study whether entity types are distributed differently over the news and social media text streams.

3. Analyzing Emerging Entities

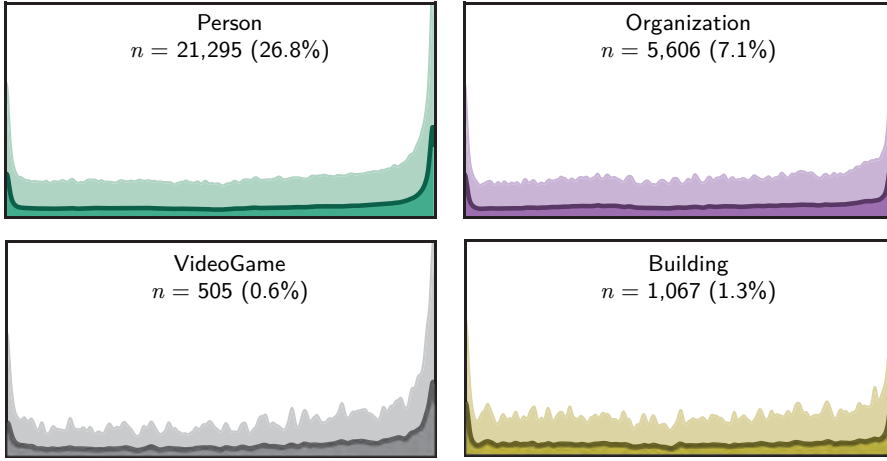


Figure 3.10: Type signatures of the Person, Organization, VideoGame and Building types. Even though the number of entities of that type differ substantially, the signatures show roughly similar patterns.

Entity Types: Temporal Patterns

First, we study each entity type in isolation, i.e., we study the descriptive statistics per entity type. Table 3.8 shows all entity types with a frequency of ≥ 400 in our dataset. We find that the entity type signatures (i.e., the average over all time series for each entity type) are highly similar to the global pattern (visualized in Figure 3.6), suggesting that the time series of mentions are highly variable within an entity type. To illustrate this, see Figure 3.10 for an example of two common entity types (top row) and two less frequently emerging types (bottom row). Whereas the signature becomes smoother as the number of entities increase, the overall pattern is highly similar across the four types.

In contrast, the descriptive statistics per entity type does yield clear patterns, as we will describe next. First, the *null* class, i.e., the entities that are not assigned an entity type in DBpedia exhibit very low emergence volumes, with an average of 98 documents over 225 days. Which may be explained by their nature, as they are not assigned a class in the DBpedia ontology, they are likely to be very long-tail, or unpopular entities.

Second, we note a group of “fast” emerging entity types, i.e., those with short emergence durations and/or high emergence velocities, e.g., *DesignedArtifact*, *CreativeWork*, *MusicalWork*, and *VideoGame*. In particular, the *DesignedArtifact* type shows high emergence velocities: it takes entities of this type on average 217 days to be incorporated into the KB, with an average volume of over 7 documents a day (versus 0.87 for the global average). The *DesignedArtifact* type includes entities such as devices and products, e.g., smartphones, tablets, and laptops. The relatively fast transition of the entities of this type may be explained by their nature: they have short “life-cycles” and may be superseded or replaced at high frequencies. Consider, e.g., the release or announcement of a new smartphone: this event typically generates a lot of attention in a short timeframe, which may result in a fast emergence. Similar to

Table 3.8: Descriptive statistics per entity type (for types that occur ≥ 400 in our dataset).

stream	n_samples	duration (#days)			volume (#docs)			velocity (docs/day)		
		mean \pm std	med.		mean \pm std	med.		mean \pm std	med.	
Person	21,295	270 \pm 151	254		243 \pm 692	71		1.03 \pm 3.32	0.32	
Athlete	8,018	260 \pm 150	235		264 \pm 674	76		1.05 \pm 2.45	0.37	
InformationEntity	7,847	242 \pm 154	210		294 \pm 1,923	90		1.42 \pm 6.53	0.51	
CreativeWork	7,795	243 \pm 154	211		294 \pm 1,928	90		1.42 \pm 6.54	0.52	
Organization	5,606	279 \pm 153	270		335 \pm 1,812	71		1.40 \pm 14.44	0.31	
Place	3,689	274 \pm 149	273		122 \pm 448	33		0.48 \pm 1.61	0.16	
Company	2,536	284 \pm 156	275		462 \pm 1,964	108		1.98 \pm 20.88	0.47	
MusicalWork	2,474	218 \pm 148	181		170 \pm 533	78		1.13 \pm 2.23	0.49	
Movie	2,033	267 \pm 154	247		279 \pm 1,322	87		1.20 \pm 6.57	0.42	
OfficeHolder	1,929	287 \pm 158	284		210 \pm 476	73		0.85 \pm 1.88	0.31	
MusicGroup	1,649	293 \pm 150	289		221 \pm 393	86		0.95 \pm 1.97	0.34	
Artist	1,624	299 \pm 152	302		240 \pm 564	75		0.95 \pm 2.19	0.30	
ArchitecturalStructure	1,591	279 \pm 149	284		133 \pm 436	36		0.47 \pm 1.14	0.18	
PopulatedPlace	1,521	262 \pm 145	244		119 \pm 481	31		0.53 \pm 2.15	0.15	
Building	1,067	281 \pm 150	290		125 \pm 374	34		0.43 \pm 0.98	0.17	
TelevisionShow	1,043	229 \pm 158	193		228 \pm 503	87		1.33 \pm 3.05	0.57	
WrittenWork	959	267 \pm 147	245		307 \pm 864	88		1.26 \pm 2.99	0.44	
EducationalInstitution	915	290 \pm 144	308		108 \pm 564	30		0.41 \pm 2.21	0.12	
Software	769	250 \pm 156	221		732 \pm 5,413	192		3.04 \pm 16.33	0.92	
School	554	280 \pm 142	305		56 \pm 158	25		0.29 \pm 2.24	0.11	
Book	524	272 \pm 147	263		286 \pm 827	93		1.17 \pm 3.03	0.44	
VideoGame	505	229 \pm 150	198		381 \pm 657	189		2.08 \pm 3.24	1.03	
DesignedArtifact	409	217 \pm 149	187		1,420 \pm 7,142	214		7.04 \pm 20.42	1.39	
Infrastructure	403	271 \pm 146	269		127 \pm 560	34		0.47 \pm 1.42	0.17	
null	39,807	225 \pm 151	196		98 \pm 861	15		0.58 \pm 3.70	0.10	

devices of the DesignedArtifact-type, creative works (CreativeWork, including MusicalWork, WrittenWork, Movie, etc.) share this characteristic: they play a central but short-lived role in public discourse.

Third, the “slower” entities, i.e., those with longer emergence durations and lower emergence volumes, are largely person types such as writers (Writer), artists (Artist), and political figures (OfficeHolder), but also schools (School and EducationalInstitution), and geographical entities (e.g., Building, ArchitecturalStructure, Place, and PopulatedPlace). These entities by their nature may have longer life-cycles, with a more gradual “rise to fame” (politicians, artists), and play a less central role in public discourse (schools, buildings). The opening of a new school may appear briefly in regional and local news sources, but is unlikely to be globally and widely reported. Politicians generally have a long and gradual career, surfacing e.g., in regional media, and do not suddenly “burst” into existence.

To better understand the difference between “fast” and “slow” entities, we examine the popularity of entities. Table 3.9 lists the average number of pageviews received per entity in 2015, grouped by type. Entity types that exhibit short emergence durations and high velocities are all in the top 10 (ranks 3, 4, and 9, for VideoGame, CreativeWork, and

3. Analyzing Emerging Entities

Table 3.9: “Popularity,” i.e., average total number of pageviews in 2015 of each entity in our dataset, aggregated per entity type. Ranked in descending order.

Rank	Type	Mean \pm std	Median
1	Movie	98,387 \pm 322,166	14,352
2	TelevisionShow	97,098 \pm 309,172	9,765
3	VideoGame	51,236 \pm 166,802	11,852
4	CreativeWork	50,024 \pm 213,490	5,716
5	InformationEntity	49,704 \pm 212,816	5,634
6	Software	43,657 \pm 149,499	9,582
7	MusicGroup	38,883 \pm 133,336	5,400
8	Artist	35,607 \pm 122,032	4,116
9	DesignedArtifact	29,830 \pm 82,081	7,191
10	Book	18,248 \pm 109,400	3,126
11	WrittenWork	14,227 \pm 86,637	1,801
12	Person	13,772 \pm 77,791	1,568
13	MusicalWork	10,443 \pm 25,009	3,523
14	Athlete	9,415 \pm 41,887	1,545
15	Organization	9,003 \pm 45,140	1,816
16	Company	7,624 \pm 21,371	2,566
17	OfficeHolder	3,763 \pm 16,167	958
18	ArchitecturalStructure	3,189 \pm 16,978	1,042
19	Building	3,180 \pm 20,106	987
20	Infrastructure	2,813 \pm 6,769	1,085
21	Place	2,339 \pm 12,649	827
22	EducationalInstitution	1,799 \pm 3,031	862
23	PopulatedPlace	1,743 \pm 9,081	694
24	School	1,137 \pm 1,426	747

DesignedArtifact, respectively), whereas the slower entity types all reside towards the lower ranks of the table, e.g., rank 19, 22, and 24 for Building, Educational-Institution and School, respectively. This suggests that entity types that emerge more quickly remain more popular over time.

Entity Types per Stream

To determine whether the entity types observed in news and social media streams differ significantly, we apply Pearson’s chi-squared test. This allows us to identify entity types that are observed more than expected (w.r.t. the global distribution) in the news and social media streams. See Table 3.10.

The majority of the entity types observed more frequently in social media streams (right column of Table 3.10) are the ones we identified to be comparatively fast in emerging in the previous section, e.g., DesignedArtifact, VideoGame, TelevisionShow, Software. The (average) popularity rankings from Table 3.9 show how 6 out of 10 of the entity types that are observed more frequently in social media are in the top 10 most

Table 3.10: Entity types ranked by exceeded observed frequency w.r.t. expected frequency using chi-grams. The popularity rank from Table 3.9 is shown in brackets.

More in news than social	More in social than news
Person (12)	DesignedArtifact (9)
Athlete (14)	VideoGame (3)
Organization (15)	Infrastructure (20)
Place (21)	Book (10)
InformationEntity (5)	TelevisionShow (2)
CreativeWork (4)	Software (6)
Company (16)	School (24)
PopulatedPlace (23)	MusicalWork (13)
OfficeHolder (17)	WrittenWork (11)
ArchitecturalStructure (18)	Movie (1)

popular entity types of Table 3.9. The average popularity rank of the social media entity types is 9.9. At the other end, the types of entities seen more frequently in news are both slower in transition on average (e.g., *Person*, *Organization*, and *Company*), and are characterized by being more “general” or less niche types compared to the entity types seen more frequently in the social media stream. Videogame or smartphones are likely to see more exposure in social media streams in, e.g., blog and forum posts, but more general entity types such as people, places, and organizations are more natural subjects for (traditional) news media. Merely 2 entity types that are more often seen in news streams are in the top 10 most popular entity types in Table 3.9. The average popularity rank of news-specific entity types is 14.5. These observations suggest that entities that remain popular over time are more likely to emerge in the social media stream.

Summary

In summary, we have shown that different entity types exhibit substantially different emergence patterns, but entities that belong to a particular type show broadly similar emergence patterns. Furthermore, we have shown that different entity types are distributed distinctly over different online text streams, which can be intuitively explained by looking at both the nature of the entity types and the nature of the streams. Next, we have seen that entities that emerge fast are more likely to remain popular over time. Finally, we have seen that in social media streams entities emerge faster, and more remain popular over time in comparison to news streams.

3.6 Conclusion

In this chapter, we studied entities as they emerge in online text streams. We did so by studying a large set of time series of mentions of entities in online news streams before they are added to Wikipedia. We studied implicit groups of similarly emerging entities by applying a burst-based agglomerative hierarchical clustering method and explicit groups

by isolating entities by whether they emerge in news or social media streams. Next, we summarize the findings, implications and limitations of our study into the nature of emerging entities in online text streams.

3.6.1 Findings

In Section 3.5.1, we applied a clustering method to the time series of mentions of emerging entities to find implicit groups of similarly emerging entities. We found that, globally, entities have a long time span between surfacing in online text streams and being incorporated into the KB. During this time span, an emerging entity is associated with multiple bursts (i.e., resurfaces into public discourse), however both the emerging entities' introduction into public discourse and subsequent incorporation into Wikipedia occur in the largest document bursts. Emergence durations and volumes show large standard deviations, indicating that they differ substantially between entities. For this reason, we turned to time series clustering to uncover distinct groups of entities. We discovered two distinct emergence patterns: early bursting (EB) entities and late bursting (LB) entities. Analysis suggests that EB entities comprise mostly "head" or popular entities; they exhibit fewer and higher bursts, with shorter emergence durations and lower emergence volumes. The LB entities emerge more slowly on average, and witness a more gradual increase of exposure in online text streams. The emergence patterns we visualized differ substantially from the global average and from, e.g., the type signatures studied in Section 3.5.3, suggesting that the entities in each of the underlying clusters exhibit substantially different and distinct emergence patterns from entities in the other clusters.

In Section 3.5.2, we showed that entities emerging in news and social media streams display very similar emergence patterns, but that on average, entities that emerge in social media have a longer period between surfacing and being incorporated into the KB. We hypothesize that this can be attributed to the nature of the underlying sources. Traditional news media is more mainstream and professional, with a larger audience and reach, and more authority than social media streams. Our findings are in line with those of Petrovic et al. [207], who compare breaking news on traditional media with that on social media. Their findings suggest reported events overlap largely between both media, however, social media exhibits in addition a long tail of minor events, which may explain the longer uptake on average. Leskovec et al. [149] find that the "attention span" for news events on social media both increases and decays at a slower rate than for traditional news sources, which may explain the comparatively slower uptake on social media.

Finally, we studied entity types in Section 3.5.3. We showed that different entity types exhibit substantially different patterns, but entities of a similar type show similar patterns. Some entity types, e.g., devices or creative works, on average emerge faster than entities such as buildings, locations, and people. At the same time, the former "faster" entity types remain more popular over time (as seen through their pageview counts). One aspect that distinguishes between "fast" and "slow" entity types, is that the former are more likely to appear in so-called "soft news" (i.e., news that covers sensational or human-interest events and topics, e.g., news related to celebrities and cultural artifacts), whereas the slower entity types are more likely to be associated with more substantive "hard news" (i.e., news that encompasses more pressing or urgent events and topics, e.g., reports related to political elections) [246]. Granka [96] studied the differences in "attention span" of the

public (as measured through search engine query volumes) and the traditional news media (as measured through coverage volume) for “hard” and “soft news,” and found that, in line with the findings of Leskovec et al. [149], hard news is associated with a relatively short period of attention from the public (as measured by query volume). Soft news exhibits a slower decrease of the public’s attention (as seen through slower declines in query volumes), which supports our finding that faster entity types — entities more likely to be associated with soft news — tend to remain more popular over time. Furthermore, the relatively longer attention for soft news may explain the quicker uptake of the “fast” entity types, e.g., “cultural” artifacts (e.g., movies, TV shows, artists) may emerge more quickly, as they are more widely supported, followed, and more strongly represented in our online public discourse.

Finally, we showed how entity types are distributed differently over news and social media streams. This difference in entity types may be explained by the nature of the streams. Partly because of the open, democratic, and user-generated nature of the Web 2.0, and blogging in particular [178], blogs no longer simply pick up news stories from the traditional media. The agenda setting power of traditional media is diluting [179]. Moreover, there are situations in which breaking news emerges on social media, e.g., the death of Osama Bin Laden [126]. And finally, as mentioned before, the events reported on social media and traditional news overlap, but social media has a long tail of its “own” events [207], which may account for the different distribution of emerging entity types in the social media stream.

Taking a step back, our findings can be summarized to the observation that emerging entities are not “born equal,” i.e., the patterns and features under which an entity emerges differ depending on source and type.

3.6.2 Implications

The findings in this chapter have implications for designing systems to detect emerging entities, and more generally for studying and understanding how entities emerge in public discourse before they are deemed important enough to be incorporated in the KB. Here, we list observations and findings that have implications for the discovery process of emerging entities. First, we have shown that entities are likely to resurface multiple times in public discourse (i.e., online text streams) before being incorporated into the KB. This means that, on average, it takes multiple moments of “exposure” for an entity to emerge. This suggests that monitoring bursts of newly emerging entities could serve as an effective method for predicting when an entity is about to be incorporated into the KB, i.e., after observing the *initial burst*, it is not too late. Furthermore, we have shown that the type of stream in which entities emerge (i.e., news and social media) carries different signals that allow us to model the process of the process of entity emergence. More specifically, we have shown how entities that emerge, e.g., in both streams at the same time tend to be added to the KB comparatively quicker than, e.g., those entities that never move from one stream to another. These findings suggest that taking different streams into account separately can be beneficial for detecting emerging entities. Finally, we have shown that the different types of online text streams also surface different types of entities. This provides insights into the difference between “mainstream media” and users of the world-wide web, in terms of preferences and interests through the “agenda setting.”

3.6.3 Limitations

The work presented in this chapter also knows several limitations.

Clusters. Part of our findings are derived from the clusters that serve as a starting point for discovering common patterns, similarities, and differences in emergence patterns. Clustering and studying cluster signatures is by design a subjective matter [254]. Applying unsupervised clustering, entailing different hyperparameters, is by definition a hard task to “evaluate,” i.e., to decide whether the resulting clusters show meaningful differences. The large (standard) deviations seen in the descriptive statistics within clusters may suggest there exists a wider variety of different entities. However, the linear interpolation step we took to cluster time series with variable lengths and different relative timespans (i.e., not temporally aligned) will result in varied time series. In our defense, the cluster signatures that result from the clustering method yielded visually discernible and statistically different patterns between clusters, which was not the case for the signatures of the groups of time series from Sections 3.5.2 and 3.5.3 (see, e.g., Figure 3.10). Finally, the clustering’s resulting dendrogram suggests there are distinct and meaningful groups of substantially different time series, as the structure of the dendrogram shows symmetry and clear separations.

Data. Next, the fragmented nature of the dataset that serves as the starting point of our study, the TREC-KBA StreamCorpus 2014, means the coverage, and hence representativeness and comprehensiveness of the data cannot be guaranteed. As could be seen in Figure 3.2, the dataset contains a blind spot around May 2012, which may affect the time series of all emerging entities. To minimize the adverse effects, we normalize all entity document mention time series by the total document volume. Furthermore, the choice of underlying sources (that represent the social media and news streams) is limited, e.g., popular social media channels such as Tumblr, Twitter and Facebook are not part of the dataset. There may be sampling bias in the types of sources, resulting in a similar bias in the entities. That is to say, with another set of sources, we may have had different findings. This is unavoidable.

The entity annotations that represent the starting point of our study into emerging entities cannot be assumed to be 100% accurate. So-called “cascading errors” [85] cause the overall accuracy to suffer, by using imperfect tagging by SERIF as input for imperfect FAKBA1 annotations. The latter annotations are estimated (from manual inspection) to contain around 9% incorrectly linked Freebase entities, with around 8% of SERIF mentions being wrongfully not linked. Even more so, the “difficult” entity links are long-tail entities, ones that are likely to be included in our filtered set, meaning the accuracy may be relatively worse in our subset of entities. However, manually correcting the annotations was beyond the scope of this study, and the large scale of the dataset makes it less likely that wrongfully linked entities are a major issue.

Finally, there may be a cultural bias inherent in our choice of datasets: we used English language news sources and social media as well as the English version of Wikipedia. As an illustration to the cultural bias inherent to the data sources, see Table 3.11 for the top 10 most frequently mentioned entities in the FAKBA1 dataset. Hence, one could claim that we studied the emergence patterns of entities for the *English speaking* part

Table 3.11: Top 10 most frequently occurring entities in the FAKBA1 dataset.

1	United States	236,705,559
2	United Kingdom	63,759,238
3	Barack Obama	61,412,413
4	China	57,273,919
5	Yahoo!	49,971,781
6	Facebook	45,602,997
7	New York City	40,043,377
8	India	39,359,865
9	Europe	36,342,113
10	Canada	29,501,709

of the world. Different datasets are likely to yield different findings. However, it is unfortunate that the English speaking part of the world is disproportionately represented in our field of research, as witnessed by the biggest constraint in conducting this study: dataset availability. We encourage the community to create suitable datasets in other languages and/or reflecting cultural practices in other parts of the planet so as to enable comparative studies.

3.6.4 What's Next?

Following our study into the nature of entities as they emerge in online text streams, and understanding that knowledge bases are dynamic in nature, in the next two chapters we propose automated methods for discovering and searching emerging entities. More specifically, in the next chapter we address the task of predicting emerging entities in social streams (Chapter 4). Next, we turn to improving their retrieval effectiveness, by constructing dynamic entity representations. More specifically, the dynamic representations aim to capture the different and dynamic ways in which people may refer to or search for entities (Chapter 5).

4

Predicting Emerging Entities

“Nec scire fas est omnia.”

—*Horace, Carmina, IV.*

4.1 Introduction

Entity linking owes a large part of its success to the extensive coverage of today’s knowledge bases; they span the majority of popular and well established entities and concepts. For most domains this broad coverage is sufficient. However, it does not provide a solid basis in domains that refer to “long-tail” entities, such as the E-Discovery domain, or in domains where new entities are constantly born, e.g., in the news and social media domain. As we have seen in the previous chapter, entities may emerge (and sometimes disappear) before editors of a knowledge base reach consensus on whether an entity should be included in the knowledge base. Knowledge bases are never complete: new entities may emerge as events unfold, but at the same time, long-tail, relatively unknown entities may also be added to a KB.

As a follow-up to Chapter 3, in this chapter we turn our attention to predicting newly emerging entities, i.e., to discover entities that appear in social media streams before they are incorporated into the KB. Identifying newly emerging entities that will be incorporated in a knowledge base is important for knowledge base construction, population, and acceleration, and finds applications in complex filtering tasks and search scenarios, where users are not just interested in finding any entity, but also in entity attributes like impact, salience, or importance. Identifying emerging entities is closely related to named-entity recognition and classification (NERC), and named-entity normalization (NEN) or disambiguation, with the additional constraint that an entity should have “impact” or be important enough to be incorporated in the knowledge base. Although impact or importance are hard to model because they depend on the context of a task or domain, we argue that entities that are included in a knowledge base are more important than those that are not, and use this signal for modeling the importance of an entity. Through this approach, our method leverages prior knowledge (as encoded in the knowledge base) for discovering newly emerging entities.

Named-entity recognition is a natural approach for identifying these newly emerging entities that are not in the knowledge base. However, current models fall short as they do

not account for the aforementioned importance or impact of the entity. In this chapter, we present an unsupervised method for generating pseudo-ground truth for training a named-entity recognizer to steer its predictions towards newly emerging entities. Our method is applicable to any trainable model for named-entity recognition. In addition, our method is not restricted to a particular class of entities, but can be trained to predict any type of entity that is in the knowledge base.

The challenge of discovering newly emerging entities is two-fold: (i) the first challenge is how to model the attribute of importance, and (ii) the second is the streaming and dynamic nature of the task, where both the input data (social media) and the updates in the knowledge base (i.e., when new entities are added) come in a stream; content that is eligible for addition in Wikipedia today, may no longer be in the future [273], which renders static training annotations unusable.

Our approach to discovering newly emerging entities in social streams answers both challenges. For the first challenge, we carefully craft the training set of a named-entity recognizer to steer it towards identifying newly emerging entities. That is, we leverage prior knowledge of the entities already in the knowledge base, to identify new entities that are likely to share the same attributes, and thus be candidates for inclusion in the knowledge base. Just as a named-entity recognizer trained solely on English person-type entities will recognize only such entities, a named-entity recognizer trained on knowledge base entities can be expected to recognize only this type of entity. For the second challenge, we provide an unsupervised method for generating pseudo-ground truth from the input stream. With this automated method we are not dependent on human annotations that are necessarily limited and domain and language specific, and newly added knowledge will be automatically included. We focus on social media streams because of the fast paced evolution of content and its unedited nature, which make it a challenging setting for predicting which entities will feature in a knowledge base. The main research question we seek to answer in this chapter is:

RQ2 Can we leverage prior knowledge of entities of interest to bootstrap the discovery of new entities of interest?

To answer this question, we propose our novel method, and formulate two sub questions. The first sub question aims to study the challenge of the noisy, unedited nature of social media streams. In order to generate high quality training data for our named-entity recognizer, we propose and experiment with two sampling methods, and answer the following subquestion:

RQ2.1 What is the utility of our sampling methods for generating pseudo-ground truth for a named-entity recognizer?

We measure utility within the task of predicting new knowledge base entities from social streams as the prediction effectiveness of a named-entity recognizer trained using our method. Next, we study the impact of the amount of prior knowledge on the effectiveness of identifying newly emerging entities. Our second subquestion is:

RQ2.2 What is the impact of the size of prior knowledge on predicting new knowledge base entities?

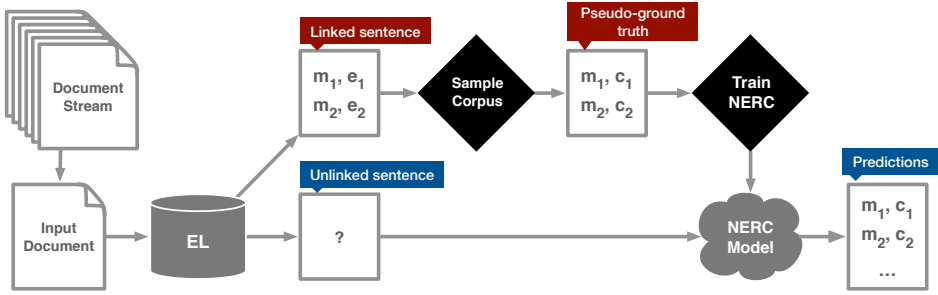


Figure 4.1: Our approach for *generating pseudo-ground truth* for training a NERC method, and for *predicting emerging entities*.

4.2 Approach

We view the task of identifying new KB entities in social streams as a combination of an *entity linking* (EL) problem and a *named-entity recognition and classification* (NERC) problem. We visualize our method in Figure 4.1. Starting from a document (social media post) in a document stream (social media stream), we extract sentences, and use an EL system to identify referent KB entities in each sentence. If any is identified, the sentence is pooled as a candidate training example for our NERC method (we refer to this type of sentence as a *linkable sentence*), otherwise it is routed to our NERC method for identifying newly emerging entities (*unlinkable sentences*): an underlying assumption behind our method is that the first place to look for emerging entities is the set of unlinkable sentences.

Most of our attention in this chapter is devoted to training the NERC method. Two ideas are important here. First, we extend the distributional hypothesis [86] (i.e., words that occur in the same contexts tend to have similar meaning) from words to entities; we hypothesize that emerging entities that should be included in the knowledge base occur in similar contexts as current knowledge base entities. Second, we apply EL on the input stream and transform its output into pseudo-ground truth for NERC; this results in an unsupervised way of generating pseudo-ground truth, with the flexibility of choosing any type of entity or concept described in the KB.

4.3 Unsupervised Generation of Pseudo-ground Truth

We start with the output of an entity linking method¹ [176, 199], on a sentence of a document in the stream. This output is our source for generating training material. The output consists of tuples of *entity mentions* and *entities* (m, e pairs in Figure 4.1).

Since we are allowed to use generic corpora from any domain, e.g., news, microblog posts, emails, chat logs, we may expect to have noise in our pseudo-ground truth. We apply various sampling methods to select sentences that make up a high quality training corpus. These sampling methods are described in Section 4.4.

¹<http://semanticize.uva.nl>

4. Predicting Emerging Entities

After sampling, we convert the remaining sentences in a format suitable for input for our NERC method. This format consists of the *entity span*; the sequence of tokens that refers to an entity, i.e. the entity mention, and *entity class* for each linked entity (m, c pairs in Figure 4.1). To denote the entity span, we apply the BIO-tagging scheme [213], where each token is tagged with whether it is the **B**eginning of an entity mention, **I**nside an entity mention, or **O**utside of an entity mention, so that a document like:

Kendrick Lamar and A\$AP Rocky. That’s when I started listening again.
Thanks to Brendan.

becomes:

Kendrick_B Lamar_I and_O A\$AP_B Rocky_I. That’s_O when_O I_O started_O
listening_O again_O. Thanks_O to_O Brendan_B.

The final step is to assign a class label to an entity. As not all knowledge bases associate classes to their entities, we use DBpedia for looking up the entity and extracting the entity’s DBpedia ontology class, if any; see Section 4.5 for details. Our example then becomes:

Kendrick_{B-PER} Lamar_{I-PER} and_O A\$AP_{B-PER} Rocky_{I-PER}. That’s_O
when_O I_O started_O listening_O again_O. Thanks_O to_O Brendan_{B-PER}.

Now we can proceed and train our NERC method with our generated pseudo-ground truth. We do so using a two-stage approach [34], where both the first stage of recognizing the entity span, and the second stage of classifying the entity type, are implemented using the fast structured perceptron algorithm [45],² which treats both stages as a sequence labeling problem.

4.4 Sampling Pseudo-ground Truth

To craft a high quality set of pseudo-ground truth for training our NERC method, in this section, we present two sampling methods: (a) sampling based on the entity linking system’s confidence score for a linked entity, and (b) sampling based on the textual quality of an input document.

4.4.1 Sampling Based on Entity Linker’s Confidence Score

Typically, entity linking systems provide confidence scores for each entity mention (n -gram) they are able to link to a knowledge base entity. These confidence scores can be used to rank possible entities for a mention, but also for pruning out mention-entity pairs for which the linker is not confident. Although the scale of the confidence score depends on the model behind the entity linking system, the scores can be normalized over the candidates for an entity mention, e.g., using linear or z-score normalization. We use the “*SenseProbability*” metric introduced by Odijk et al. [199] as our confidence score.

²<https://github.com/larsmans/seqlearn>

Table 4.1: Features used for sampling documents from which we train a NERC system.

Feature	Description	Feature	Description
n_mentions	No. of usernames (@)	avg_token_len	Average token length
n_hashtags	No. of hashtags (#)	tweet_len	Length of tweet (char)
n_urls	Number of URLs	density	Density as in [125]
ratio_upper	Percentage of upper-cased chars	personal	Contains personal pronouns (I, me, we, etc.)
ratio_nonalpha	Percentage of non-alphanumeric chars		

SenseProbability is calculated by estimating the probability of an n -gram being used as an anchor text pointing to a specific entity e , with the prior probability of the n -gram n being used as an anchor at all (als known as the *commonness*-score [182]. See equation 4.1.

$$SenseProbability(n, e) = \frac{|L_{n,e}|}{\sum_{e' \in KB} c(n, e')}, \quad (4.1)$$

where $L_{n,e}$ denotes the set of all links with n -gram n and target entity e , n denotes the n -gram to be linked (i.e., anchor text), e denotes the candidate entity, $c(n, e)$ denotes the number of times that n -gram n links to candidate entity e .

4.4.2 Sampling Based on Textual Quality

Taking the textual quality of content into account has proved helpful in a range of tasks. Based on [170, 258], we consider nine features indicative of textual quality; see Table 4.1. While not exhaustive, our feature set is primarily aimed at social streams as our target document stream (see Section 4.5) and suffices for providing evidence on whether this type of sampling is helpful for our purposes. Based on these features, we compute a final score for each document d as

$$score(d) = \frac{1}{|F|} \sum_{f \in F} \frac{f(d)}{\max_f}, \quad (4.2)$$

where F is our set of feature functions (Table 4.1) and \max_f is the maximum value of f we have seen so far in the stream of documents. Since all features are normalized in $[0, 1]$, $score(d)$ has this same range. As a qualitative check, we rank documents from the MSM2013 [19] dataset using our quality sampling method and list the top-5 and bottom-5 scoring documents in Table 4.2. Top scoring documents are longer and denser in information than low scoring documents. We assume that these documents are better examples for training a NERC system.

In the next section, we follow a linear search approach to sampling training examples as input for NERC. First, we find an optimal threshold for confidence scores, and fix it. For sampling based on textual quality, we turn to the MSM2013 dataset to determine sampling thresholds. We calculate the scores for each tweet, and scale them to fall between $[0, 1]$. We then plot the distribution of scores, and bin this distribution in three parts: tweets

4. Predicting Emerging Entities

Table 4.2: Ranking of documents in the MSM2013 dataset based on our quality sampling method. Top ranking documents appear longer and denser in information than low ranking documents.

Top-5 quality documents
<i>“Watching the History channel, Hitler’s Family. Hitler hid his true family heritage, while others had to measure up to Aryan purity.”</i>
<i>“When you sense yourself becoming negative, stop and consider what it would mean to apply that negative energy in the opposite direction.”</i>
<i>“So. After school tomorrow, french revision class. Tuesday, Drama rehearsal and then at 8, cricket training. Wednesday, Drama. Thursday ... (c)”</i>
<i>These late spectacles were about as representative of the real West as porn movies are of the pizza delivery business Que LOL</i>
<i>Sudan’s split and emergence of an independent nation has politico-strategic significance. No African watcher should ignore this.</i>
Bottom-5 quality documents
<i>Toni Braxton ~ He Wasnt Man Enough for Me _HASHTAG_ _HASHTAG_? _URL_ RT _Mention_</i>
<i>“tell me what u think The GetMore Girls, Part One _URL_”</i>
<i>this girl better not go off on me rt</i>
<i>“you done know its funky! – Bill Withers” “Kissing My Love”” _URL_ via _Mention_”</i>
<i>This is great: _URL_ via _URL_</i>

that fall within a single standard deviation of the mean are considered *normal*, tweets to the left of this bin are considered *noisy*, whilst the remaining tweets to the right of the distribution are considered *nice*. We repeat this process for our tweet corpus, using the bin thresholds gleaned from the MSM2013 set.

4.5 Experimental Setup

In addressing the problem of predicting emerging entities in social media streams, we concentrate on developing an unsupervised method for generating pseudo-ground truth for a NERC method, and predicting newly emerging entities. In particular, we want to know the effectiveness of our unsupervised pseudo-ground truth (UPGT) method over a random baseline and a lexical matching baseline, and the impact on effectiveness of our two sampling methods. To answer these questions, we conduct both optimization and prediction experiments.

4.5.1 Dataset

As our document stream, we use tweets from the TREC 2011 Microblog dataset [233], a collection of 4,832,838 unique English Twitter posts. This choice is motivated by the unedited and noisy nature of tweets, which can be challenging for prediction. Our knowledge base (KB) is a subset of Wikipedia from January 4, 2012, restricted to entities that correspond to the NERC classes person (PER), location (LOC), or organization (ORG). We use DBpedia to perform selection, and map the DBpedia classes *Organisation*, *Company*, and *Non-ProfitOrganisation* to ORG, *Place*, *PopulatedPlace*, *City*, and *Country* to LOC, and *Person* to PER.³ Our final KB contains 1,530,501 entities.

4.5.2 Experiment I: Sampling Pseudo-ground Truth

To study the utility of our sampling methods, we turn to the impact of setting a threshold on the entity linking system’s confidence score (Experiment Ia.) and the effectiveness of our textual quality sampling (Experiment Ib.).

In Experiment Ia., we perform a sweep over thresholds between 0 and 1, in steps of 0.1, using the same threshold for both the generation of pseudo-ground truth and evaluating the prediction effectiveness of emerging entities. Lower thresholds allow low confidence entities in the pseudo-ground truth, and likely generate more data at the expense of noisy output. We emphasize that we are not interested in the correlation between noise and confidence score, but rather in the performance of finding emerging entities given the entity linking system’s configuration. In Experiment Ib., we compare our methods performance with differently sampled pseudo-ground truths, i.e., by sampling tweets that belong to either the *nice*, *normal*, or *noisy* bins, following our textual quality sampling method (cf. Section 4.4.2).

4.5.3 Experiment II: Prediction Experiments

To answer our second research question, and study the impact of prior knowledge on detecting emerging entities, we compare the performance of our method (UPGT) to two baselines: a *random baseline* (RB) that extracts all n -grams from test tweets and considers them emerging entities, and a *lexical-matching baseline* (NB) that follows our approach, but generates pseudo-ground truth by applying lexical matching of entity titles, instead of an EL system, and refrains from sampling based on textual quality. For this experiment, we use the optimal sampling parameters for generating pseudo-ground truth from our previous experiment, i.e., include linked entities with a confidence score higher than 0.7, and use only normal tweets. As we will show in Section 4.6, the threshold of 0.7 balances performance with high recall of entities in the pseudo-ground truth.

4.5.4 Evaluation

We evaluate the quality of the generated pseudo-ground truth on the effectiveness of a NERC system trained to predict newly emerging entities. As measuring the addition of new entities to the knowledge base is non-trivial, we consider a retrospective scenario: Given

³<http://mappings.dbpedia.org/server/ontology/classes/>

our KB, we randomly sample entities to yield a smaller KB (KB_s). This KB_s simulates the available knowledge at the present point in time, whilst the full KB represents the future state. By measuring how many entities we are able to detect in our corpus that feature in KB, but not KB_s , we can approximate the newly emerging entity prediction task. We create KB_s by taking random samples of 20–90% the size of KB (measured in number of entities), in steps of 10%. We repeat each sampling step ten times to avoid bias.

For each KB_s , we generate pseudo-ground truth for training and test sets for evaluation. We use KB_s to link the corpus of tweets, and yield two sets of tweets: (a) tweets with linked entities, (b) tweets that may contain emerging entities, analog to the *linked* and *unlinked sentences* from Figure 4.1. The size of these two sets depends on the size of KB_s : a smaller KB will yield a larger set of unlinked tweets. This makes comparisons of results across different KB_s difficult. We cater for this bias by randomly sampling 10,000 tweets from both the test set and the pseudo-ground truth, and repeating our experiments ten times. Using the smallest KB_s (20%) results in about 15,000 tweets in the pseudo-ground truth. Ground truth is assembled by linking the corpus of tweets using the full KB. The ground truth consists of 82,305 tweets, with 12,488 unique entities.

We evaluate the effectiveness of our method in two ways: (a) the ability of NERC to generalize from our pseudo-ground truth, and (b) the accuracy of our predictions.

For the first, we compare the predicted *entity mentions* to those in our ground truth, akin to traditional NERC evaluation. For the second we take the set of correctly predicted entity mentions (true positives), and link them to their referent entities in the ground truth. This allows us to measure what we are actually interested in: the fraction of newly discovered entities. For both types of evaluation we report on average precision and recall over 100 runs per KB_s . Statistical significance is tested using a two-tailed paired t-test and is marked as \blacktriangle for significant differences for $\alpha = .01$.

4.6 Results

4.6.1 Experiment I: Sampling Pseudo-ground Truth

Our first experiment aims to answer RQ2.1:

RQ2.1 What is the utility of our sampling methods for generating pseudo-ground truth for a named entity recognizer?

For this experiment, we fix the size of the KB_s at 50%. We start by looking at the ability of NERC to generalize from our pseudo-ground truth, measured on two aspects: (i) effectiveness for identifying mentions of newly emerging entities (i.e., *entity mentions*), and (ii) predicting newly emerging entities (i.e., *entities*).

In Experiment Ia., we look at our confidence-based sampling method. For identifying mentions of newly emerging entities, we find that effectiveness peaks at 0.1 confidence threshold with a precision of 38.84%, dips at 0.2 and slowly picks up to 35.75% as threshold increases (Figure 4.2, left). For identifying the newly emerging entities that are referred to by the mentions, effectiveness positively correlates with the threshold. Effectiveness peaks at the 0.8 confidence threshold, statistically significantly different from 0.7 but not from 0.9 (Figure 4.2, right).

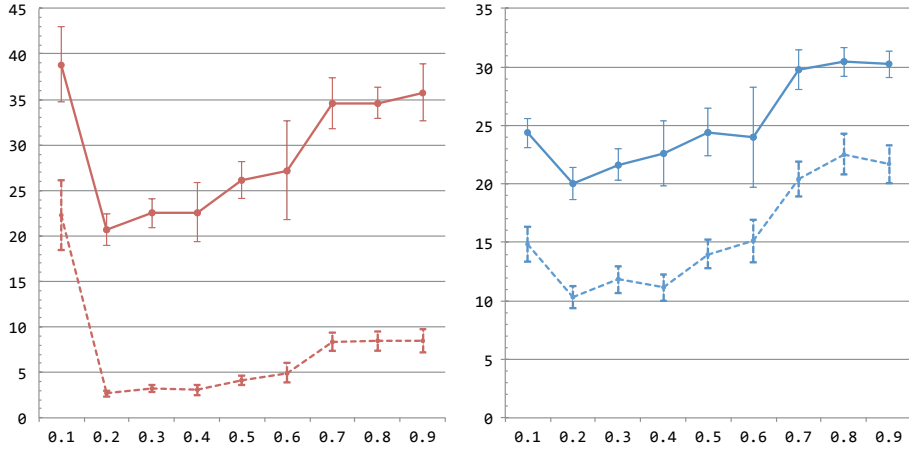


Figure 4.2: **Experiment Ia.** Impact of confidence score on UPGT. Effectiveness of identifying entity *mentions* is shown in the left plot, effectiveness of identifying newly emerging *entities* in the right. Threshold set on the confidence score on the x-axis. Precision (solid line) and recall (dotted) are shown on the y-axis.

Table 4.3: Number of predicted emerging entity mentions (P) per threshold on the confidence score (T). GT = Ground truth.

T	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
P	6,653	1,500	1,618	1,512	1,738	2,025	2,662	2,713	2,614
GT	11,429	11,533	11,291	11,078	10,955	10,935	10,799	10,881	10,855

Interestingly, besides precision, recall also shows a positive correlation with thresholds. This suggests that in identifying newly emerging entities, missing training labels are likely to have less impact on performance than generating incorrect, noisy labels. This is an interesting finding as it sets the emerging entity prediction-task apart from traditional NERC, where low recall due to incomplete labeling is a well-understood challenge.

Next, we turn to the characteristics of the pseudo-ground truth that results for each of these thresholds, and provide an analysis of their potential impact on effectiveness. We find that more data through a larger pseudo-ground truth allows NERC to better generalize and predict a larger number of emerging entities. This claim is supported by the number of predicted entity mentions per threshold in Table 4.3. We find a similar trend as in the precision and recall graph above: the number of predicted emerging entities peaks for the threshold at 0.1 (6,653 entity mentions), and drops between 0.2 and 0.4, and picks up again from 0.5 reaching another local maximum at 0.8. The increasing number of predicted emerging entity mentions with stricter thresholds indicates that the NERC model is more successful in learning patterns for separating entities from noisy labels. This may be due to the entity linking system linking only those entities it is most confident about, providing a clearer training signal for NERC.

For the rest of our experiments we use a threshold of 0.7 on confidence score because

4. Predicting Emerging Entities

Table 4.4: **Experiment Ib.** Precision and recall for three sampling strategies based on textual quality of documents: nice, normal, normal+nice (mixed). We also report on effectiveness of not using sampling by textual quality for reference. Boldface indicates best performance. Statistical significance is tested against the previous sampling method, e.g., nice to normal.

Sampling	Mention		Entity	
	Precision	Recall	Precision	Recall
No sampling	34.26±2.65	8.21±0.83	29.63±1.67	20.25±1.56
Normal+nice	45.50±4.71 [▲]	12.97±2.03 [▲]	36.22±2.07 [▲]	24.86±1.69 [▲]
Normal	66.09±3.86 [▲]	30.94±3.46 [▲]	44.62±1.51 [▲]	32.20±1.67[▲]
Nice	70.36±3.07[▲]	30.98±3.25	45.99±1.34[▲]	29.69±1.79

it is deemed optimal in terms of trade-off between performance and quantity of entities in pseudo-ground truth.

In Experiment Ib., we study three different textual quality-based sampling strategies: we consider only tweets that fall in the *normal* bin (i), tweets that fall in the *nice* bin (ii), or tweets that fall in both the *normal* and *nice* bins (iii); see Section 4.4. For reference, we also report on the performance achieved when no textual quality sampling is used. We keep KB_S fixed at 50%, and use 0.7 for confidence threshold.

Textual quality-based sampling turns out to be twice as effective as no sampling on both identifying mentions of emerging entities, and identifying newly emerging entities. Among our sampling strategies, nice proves to be the most effective with a precision of 70.36% for entity mention identification. In emerging entity prediction, the performance of nice and normal strategies hovers around the same levels. In terms of recall, nice and normal methods are on par, outperforming both other strategies. The success of nice and normal sampling methods can be attributed to the fact that a more coherent and homogeneous training corpus allows the NERC model to more easily learn patterns.

4.6.2 Experiment II: Impact of Prior Knowledge

Next, we seek to study the impact of the size of the prior knowledge that is available to our emerging entity prediction method. We do so by answering RQ2.2:

RQ2.2 What is the impact of the size of prior knowledge on predicting newly emerging entities?

We use the optimal combination of our sampling methods from the previous experiments, i.e., a confidence threshold of 0.7, and the normal textual quality sampling. We again look at the effectiveness of our methods in both identifying emerging mentions, and emerging entities.

Figure 4.3 shows the effectiveness of our methods as a function of the size of the knowledge base (i.e., size of KB_S). For identifying mentions of emerging entities, our method (UPGT, blue line) constantly and statistically significantly outperforms both lexical (red line) and random baselines (not shown). In terms of recall, the lexical baseline

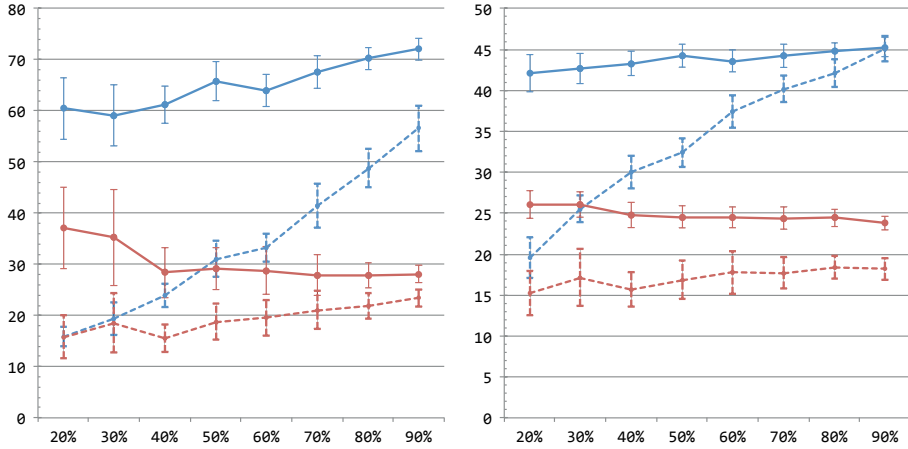


Figure 4.3: Our method (UPGT, blue line) versus the lexical baseline (red) for both identifying mentions of emerging entities (left) and newly emerging entity detection (right). Knowledge base size is on the x -axis, and precision (solid lines) and recall (dotted lines) are marked on the y -axis.

is on par with UPGT for KB sizes up to 30%. The random baseline shows very low precision for both identifying mentions of emerging entities, and predicting emerging entities over all KBs, but almost perfect recall—which is expected given that it assigns all possible n -grams as emerging entity mentions and newly emerging entities (0.69% precision and 65% recall for entity mention identification, 1.82% precision and 94.95% recall for emerging entity prediction).

Next, we take a closer look at the results. The lexical baseline’s recall increases slightly when more prior knowledge is added to the knowledge base. This is expected behavior because, as we saw in our previous experiment, the pseudo-ground truth gets more diverse labels, and helps the NERC to generalize. The number of unique entities in the pseudo-ground truth shows that the number increases with the size of KBs. UPGT assigns labels to 2,500 unique entities at 20% KBs, which tops at 11,000 unique entities at 90%. These numbers are lower for the lexical baseline (1,800 at 20% KBs, and 7,000 at 90% KBs). However, for both methods, the number of entities in the ground truth stays around the same. The gradual improvement in precision and recall of UPGT for increasing KBs can be attributed to a broader coverage for labeling (observed through looking at the prior entities), and the main distinction between the lexical baseline and UPGT: stricter labeling through leveraging the entity linking system’s confidence score.

Finally, to better understand the performance of our method, we have looked at the set of correct predictions (emerging entities), and false positives, or incorrectly identified emerging entities. Our analysis revealed examples of actual emerging entities, i.e., entities that were not included in the initial KB, but did exist in the Wikipedia dump. This may be attributed due to, e.g., missing DBpedia class labels, which highlights the challenging setting of evaluating the task.

On the whole, however, our method is able to deal with missing labels and incomplete

data, as observed through its consistent and stable precision, justifying our assumption that data is incomplete by design.

4.7 Conclusion

In this chapter, we tackled the problem of predicting emerging entities in social streams, and presented an effective method of leveraging prior knowledge to bootstrap the discovery of new entities of interest. We set out to answer the following question:

RQ2 Can we leverage prior knowledge of entities of interest to bootstrap the discovery of new entities of interest?

We presented an unsupervised method for generating pseudo-ground truth using an EL method with a reference KB. The pseudo-ground truth serves as training data for a NERC method for detecting emerging entities that are likely to be incorporated in a KB. To answer RQ2 we formulated and answered two subquestions. In Section 4.6.1 we answer our first subquestion :

RQ2.1 What is the utility of our sampling methods for generating pseudo-ground truth for a named-entity recognizer?

We do so by introducing and studying two different sampling methods. The first sampling method is based on the entity linking system’s confidence score, where we hypothesize that keeping only high-confidence links will yield less training data but of higher quality, resulting in better predictions. The second sampling method is based on the textual quality of the input documents. We find that sampling by textual quality improves performance of NERC and consequently our method’s performance in predicting emerging entities. As setting a higher threshold on the entity linking system’s confidence score for generating pseudo-ground truth results in fewer labels but better performance, we show that the NERC is better able to separate noise from entities that are worth including in a KB. The entity linker’s confidence score is an effective signal for this separation. Both our sampling methods significantly improve emerging entity prediction.

Next, in Section 4.6.2, we answer our second subquestion:

RQ2.2 What is the impact of the size of prior knowledge on predicting new KB entities?

We do so by studying the impact of differently sized (seed) knowledge bases, i.e., different amounts of prior knowledge on the entities of interest. We find that in the case of a small amount of prior knowledge, i.e., limited size of the available initial knowledge, our method is able to cope with missing labels and incomplete data, as observed through its consistent and stable precision. This finding justifies our proposed method that assumes incomplete data by design. Furthermore, this finding suggests the scenario of an increasing rate of emerging entity prediction, as more data is fed back to the KB. Additionally, we found that a larger number of entities in the KB allows for setting a desirable stricter threshold on the confidence scores, and leads to improvements in both precision and recall. This finding suggests an adaptive threshold that takes prior knowledge into account could prove effective.

In summary, we have shown that we can effectively leverage prior knowledge (through a reference KB) to detect similar entities that are not in the KB. An implication of this finding in the discovery context is that this method allows to effectively support the exploratory search process, as it successfully identifies entities that are similar to a set of seed entities of interest.

Our work has several limitations. We studied the emerging entity prediction task in a streaming scenario. However, due to limitations of the availability of suitable datasets, we employ a retrospective scenario in Experiment II (Section 4.5.3), where we randomly sample entities from the KB for measuring the impact of the amount of prior knowledge on prediction performance. Taking this retrospective scenario, makes it impossible to measure the impact of *novelty* of entities as they emerge. By repeating each experiment ten-fold, we alleviate the problem of keeping or removing popular head entities that impact performance more than tail entities, but a more realistic scenario, e.g., using a comprehensive archive of tweets, spanning multiple months, would allow us to sample KB by time. This scenario, similar to the scenario we explored in Chapter 3 would be able to provide more insights into the relation between social media dynamics and prediction accuracy of emerging entities.

A natural extension to the work presented in this chapter would be to include a subsequent entity clustering or disambiguation step, and ultimately to “feed back” emerging entities to the KB. One approach could be to adapt the work of creating keyphrase-based representation as seed representations [120]. However, as we have seen in Chapter 3, entities are not static, and may appear in sudden bursts of documents, e.g., through a peak in interest as real-world events unfold. In the next chapter, we address this dynamic nature of entities, in the context of entity retrieval. We propose a method for dynamically constructing entity representations by leveraging the way in which people refer to entities online, to improve the retrieval effectiveness of entities.

5

Retrieving Emerging Entities

“En wat zijn mijn woorden waard als ik ze niet meer weeg?”

—*Sticks, Waar Wacht Je Op?*

5.1 Introduction

In the third and final chapter of this first part of the thesis we focus on *retrieving* entities of interest from the knowledge base. As entities play a central role in exploratory search and answering the 5 W's [268], ranking entities to user-issued search queries is an important, but challenging task. In this chapter, we study *entity ranking*, where the goal is to position a relevant entity from the knowledge base at the top of the ranking for a given query.

We have seen in Chapter 3 that entities may suddenly appear in public discourse, e.g., through events as they unfold in the real world. Due to the dynamic nature of entities in public discourse, the way in which people address or refer to entities may suddenly change, providing challenges for traditional retrieval methods that rely on static representations. Here, we address the dynamic nature of entities in the general web search scenario, in which a searcher enters a query that can be satisfied by returning an entity from Wikipedia. We propose a novel method that collects information from external sources and dynamically constructs entity representations, by optimally combining different entity descriptions from external sources into a single entity representation. The method learns directly from users' past interactions (i.e., searches) to adapt the entity representation towards improved retrieval effectiveness of entities.

The general web search engine scenario is motivated by the fact that many queries issued to general web search engines are related to entities [145]. Entity ranking is therefore becoming an ever more important task [13, 14, 57–59]. Entity ranking is inherently difficult due to the potential mismatch between the entity's description in a knowledge base and the way people refer to the same entity when searching for it. When we look at how entities are described, two aspects, context and time, are of particular interest and pose challenges to any solution to entity ranking. Here we explain both aspects.

Context dependency: Consider the entity *Germany*. A history student could expect this entity to show up when searching for entities related to World War II. In contrast, a sports fan searching for World Cup 2014 soccer results is also expecting to find the



Figure 5.1: Entity description of *Anthropornis* in Wikipedia and a tweet with an alternative description of the same entity.

same entity. The challenge then becomes how to capture these different contexts for one single entity.

Time dependency: Entities are not static in how they are perceived. Consider *Ferguson, Missouri*, which had a fairly standard city description before the shooting of Michael Brown happened in August 2014. After this event, the entity description of *Ferguson* changed substantially, reflecting people’s interest in the event, its aftermath and impact on the city.

We propose a method that addresses both challenges raised above. First, we use the collective intelligence as offered by a wide range of entity “description sources” (e.g., tweets and tags that mention entities), and we combine these into a “collective entity representation,” i.e., a representation that encapsulates different ways of how people refer to or talk about the entity. Consider the example in Figure 5.1 in which a tweet offers a very different way to refer to the entity *Anthropornis* than the original knowledge base description does.

Second, our method takes care of the time dependency by incorporating dynamic entity description sources, which in turn affect the entity descriptions in near real time. Dynamics is part of our method in two ways: (i) we leverage dynamic description sources to expand entity representations, and (ii) we learn how to combine the different entity descriptions for optimal retrieval at specific time intervals. The resulting *dynamic collective entity representations* capture both the different contexts of an entity and its changes over time. We refer to the collection of descriptions from different sources that are associated with an entity as the entity’s *representation*. Our method is meant to construct the optimal

representation for retrieval, by assigning weights to the descriptions from different sources.

Collecting terms associated with documents and adding them to the document (document expansion) is in itself not a novel idea. Previous work has shown that it improves retrieval effectiveness in a variety of retrieval tasks such as speech retrieval [232] and ad-hoc web search [73, 180, 259, 270]. However, while information retrieval on the open web is inherently dynamic, document expansion for search has mainly been studied in static, context-independent settings, in which expansion terms from one source are aggregated and added to the collection before indexing [147, 180, 185, 259]. In contrast, our method leverages different dynamic description sources (e.g., queries, social media, web pages), and in this way, uses collective intelligence to bridge the gap between the terms used in a knowledge base’s entity descriptions and the terms that people use to refer to entities.

To achieve this, we represent entities as fielded documents [162], where each field contains content that comes from a single description source. In a dynamic setting such as our entity ranking setting, where new entity descriptions come in as a stream, learning weights for the various fields in batch is not optimal. Ideally, the ranker continuously updates its ranking model to successfully rank the entities and incorporate newly incoming descriptions. Hence, constructing a dynamic entity representation for optimal retrieval effectiveness boils down to dynamically learning to optimally weight the entity’s fields that hold content from the different description sources. To this end we exploit implicit user feedback (i.e., clicks) to retrain our model and continually adjust the weights associated to the entity’s fields, much like online learning to rank [121].

Our dynamic collective entity representations generate one additional challenge, which is related to the heterogeneity that exists among entities and among description sources. Popular head entities are likely to receive a larger number of external descriptions than tail entities. At the same time the description sources differ along several dimensions (e.g., volume, quality, novelty). Given this heterogeneity, linearly combining retrieval scores (as is commonly done in structured retrieval models) proves to be suboptimal. We therefore extend our method to include features that enable the ranker to distinguish between different types of entities and stages of entity representations. The main research question we seek to answer in this chapter is:

RQ3 Can we leverage collective intelligence to construct entity representations for increased retrieval effectiveness of entities of interest?

To answer this question, we formulate and seek to answer the following three subquestions. First, we check the underlying assumption of our method, and answer our first research question:

RQ3.1 Does entity ranking effectiveness increase by using dynamic collective entity representations?

To answer this question, we compare a baseline entity ranking method based on information in the knowledge base only to our method that incorporates additional description sources (web anchors, queries, tags, and tweets).

Next, we extend our method, and study the contribution of informing the entity ranking of the entity’s description state. We seek to answer the second subquestion:

RQ3.2 Does entity ranking effectiveness increase when employing additional field and entity importance features?

We answer this question by incorporating field and entity importance features to our knowledge base-only baseline ranker and our proposed method, and compare their performance. Finally, we answer this chapter’s third research question, by studying the dynamic aspect of entity ranking:

RQ3.3 Does entity ranking effectiveness increase when we continuously learn the optimal way to combine the content from different description sources?

We compare a static entity ranking baseline that is trained once at the start, to our proposed method that is retrained at regular intervals.

The main contribution of this chapter is a novel approach to constructing dynamic collective entity representations, which takes the temporal and contextual dependencies of entity descriptions into account. We show that dynamic collective entity representations better capture how people search for entities than their original knowledge base descriptions. In addition, we show how field importance features better inform the ranker, thus increasing retrieval effectiveness. Furthermore, we show how continuously updating the ranker enables higher ranking effectiveness. Finally, we perform extensive analyses of our results and show that incorporating dynamic signals into the dynamic collective entity representation enables a better matching of users’ queries to entities.

5.2 Dynamic Collective Entity Representations

5.2.1 Problem Statement

The problem of entity ranking is: given a query q and a knowledge base KB populated with entities $e \in KB$, find the best matching e that satisfies q . Both e and q are represented in some (individual or joint) feature space that captures a range of dimensions, which characterize them individually (e.g., content, quality) and jointly (e.g., relationships through click logs).

The entity ranking problem itself is a standard information retrieval problem, where the system needs to bridge the gap between the vocabulary used by users in queries, and the vocabulary of the entity descriptions. This is a long-standing but still open problem that has been tackled from many perspectives. One is to design better similarity functions, another is to develop methods for enhancing the feature spaces. Our method shares characteristics with both perspectives, as we will now explain.

5.2.2 Approach

Our approach to the entity ranking problem consists of two interleaved steps. First, we use external description sources (described in Section 5.2.3) to expand entity representations and reduce the vocabulary gap between queries and entities. We do so by representing entities as fielded documents, where each field corresponds to content that comes from one description source. Second, we train a classification-based entity ranker, that employs

different types of features to learn to weight and combine the content from each field of the entity for optimal retrieval (Section 5.2.4).

External description sources may continually update and change the content in the entity’s fields, through user feedback (i.e., clicks) following an issued query, or when users generate content on external description sources that is linked to a KB entity (e.g., Twitter, Delicious). Consequently, the feature values that represent the entities change, which may invalidate previously learned optimal feature weights and asks for continuously updating the ranking model.

5.2.3 Description Sources

To construct dynamic collective entity representations we use two types of description sources: (i) a knowledge base (*KB*) from which we extract the initial entity representations, and (ii) a set of external description sources that we use to expand the aforementioned entity representation.

Although in theory all types of external sources are allowed in the model, choosing sources that provide short text descriptions for an entity is favorable as they inherently “summarize” entities to a few keywords and do not require an additional step for filtering salient keywords. Figure 5.2 provides an example of expanding the representation of the entity Tupac Shakur¹ from several sources, each contributing somewhat different keywords.

We differentiate between external description sources that are non-timestamped (*static*) and ones that are timestamped (*dynamic*). Non-timestamped sources are those where no time information is available, and sources that are not inherently dynamic, e.g., web archives and aggregates over archives like web anchors. Timestamped external description sources are sources whose content is associated with a timestamp, and where the nature of the source is inherently dynamic or time-dependent, e.g., tweets or query logs. We describe each type of external description source below, the number of total descriptions, and affected entities, and we provide a summary in Table 5.1.

Initial Entity Representation

Knowledge base. The knowledge base that we use as our initial index of entities, and which we use to construct the initial entity representations, is a snapshot of Wikipedia from August 3, 2014 with 14,753,852 pages. We filter out non-entity pages (“special” pages such as category, file, and discussion pages), yielding 4,898,356 unique entities. The initial entity representations consist of the title and body (i.e., article content) of the Wikipedia page.

Static Description Sources

Knowledge base. Knowledge base entities have rich metadata that can be leveraged for improving retrieval [12, 226, 230]. We consider four types of metadata to construct the KB entity representations: (i) anchor text of inter-knowledge base hyperlinks, (ii) redirects, (iii) category titles, and (iv) titles of entities that are linked from and

¹https://en.wikipedia.org/wiki/Tupac_Shakur

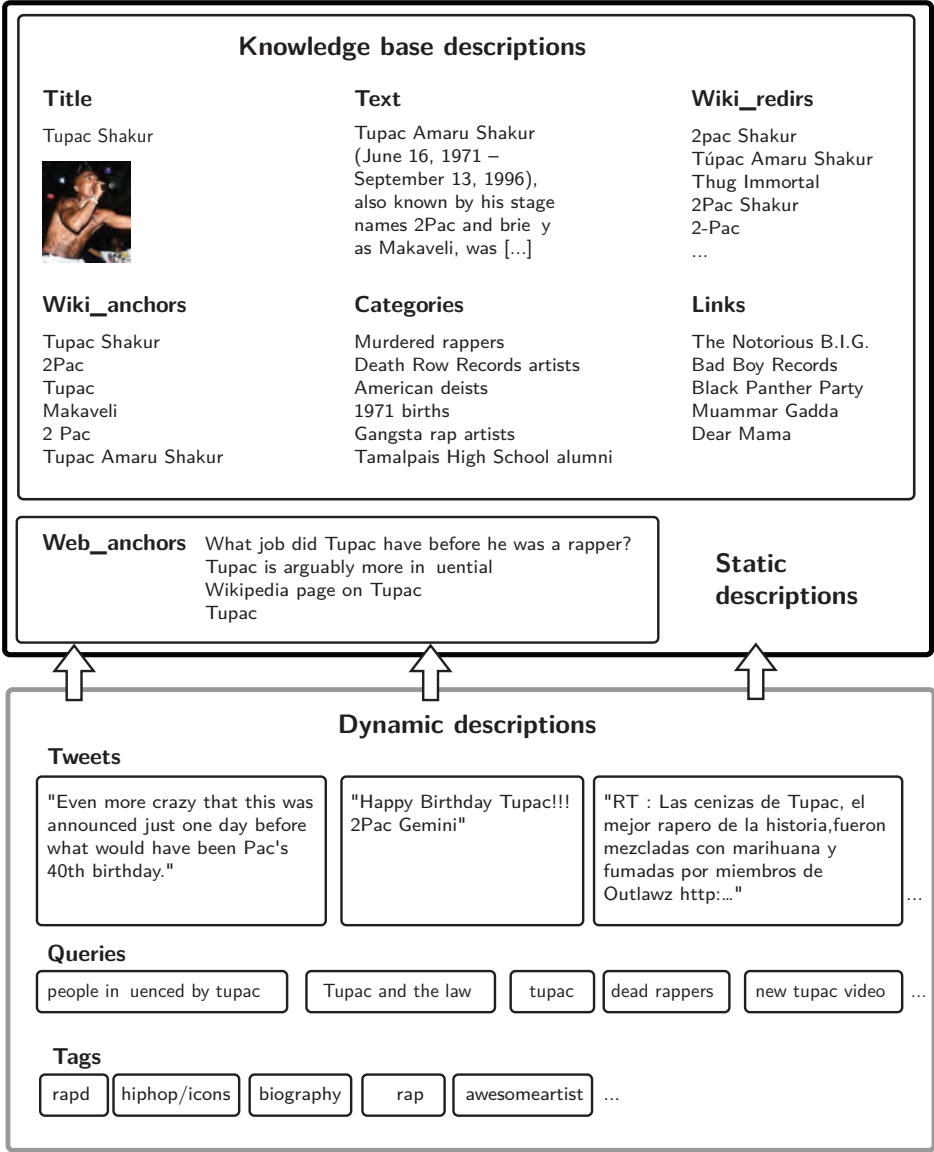


Figure 5.2: Example expansions for the entity Tupac Shakur.

to each entity. Editorial conventions and Wikipedia’s quality control ensures these expansions to be of high quality.

Web anchors. Moving away from the knowledge base itself, the web provides rich information on how people refer to entities leading to tangible improvements in retrieval [259]. We extract anchor texts of links to Wikipedia pages from the

Table 5.1: Summary of the nine description sources we consider: Knowledge Base entity descriptions (KB), KB anchors, KB redirects, KB category titles, KB inter-hyperlinks, queries, web anchors, tweets, and tags from Delicious.

Data source	Size	Period	Affected entities
<i>Static expansion sources</i>			
KB	4,898,356	August 2014	–
KB anchors	15,485,915	August 2014	4,361,608
KB redirects	6,256,912	August 2014	N/A
KB categories	1,100,723	August 2014	N/A
KB inter-links	28,825,849	August 2014	4,322,703
<i>Dynamic expansion sources</i>			
Queries	47,002	May 2006	18,724
Web anchors	9,818,004	2012	876,063
Twitter	52,631	2011–2014	38,269
Delicious	4,429,692	2003–2011	289,015

Google Wikilinks corpus.² We collect 9,818,004 anchor texts for 876,063 entities. Web anchors differ from KB anchors as they can be of lower quality (due to the absence of editorial conventions) but also of much larger volume. While in theory web anchors could be associated with timestamps, in a typical scenario they are aggregated over large archives, where extracting timestamps for diverse web-pages is non-trivial.

Dynamic Description Sources

Twitter. Mishne and Lin [185] show how leveraging terms from tweets that do not exist in the pages linked to from tweets can improve retrieval effectiveness of those pages. We follow a similar approach and mine all English tweets that contain links to Wikipedia pages that represent the entities in our KB. These are extracted from an archive of Twitter’s sample stream, spanning four years (2011–2014), resulting in 52,631 tweets for 38,269 entities.

Delicious. Social tags are concise references to entities and have shown to outperform anchors in several retrieval tasks [194]. We extract tags associated with Wikipedia pages from SocialBM0311³ resulting in 4,429,692 timestamped tags for 289,015 entities.

Queries. We use a publicly available query log from MSN sampled between May 1 and May 31, 2006, consisting of 15M queries and their metadata: timestamps and URLs of clicked documents. We keep only queries that result in clicks on Wikipedia pages that exist in our snapshot of Wikipedia, resulting in 47,002 queries associated with 18,724 uniquely clicked entities. We hold out 30% of the queries for development

²<https://code.google.com/p/wiki-links/>

³<http://www.zubiaga.org/datasets/socialbm0311/>

(e.g., parameter tuning, feature engineering; 14,101 queries) and use 70% (32,901 queries) for testing. Our use of queries is twofold: (i) queries are used as input to evaluate the performance of our entity ranking approach, and also (ii) as external description source, to expand the entity description with the terms from a query that yield a click on an entity. While this dual role of queries may promote head entities that are often searched, we note that “headness” of entities differs across description sources, and even tail entities may benefit from external descriptions (as illustrated by the *Anthropornis* example in Fig. 5.1).

5.2.4 Adaptive Entity Ranking

The second step in our method is to employ a supervised entity ranker that learns to weight the fields that hold content from the different description sources for optimal retrieval effectiveness. Two challenges arise in constructing collective dynamic entity representations:

Heterogeneity. External description sources exhibit different dynamics in terms of volume and quality of content [152], and differences in number and type of entities to which they link (see, e.g., Table 5.1). This heterogeneity causes issues both within entities, since different description sources contribute different amounts and types of content to the entity’s fields, and between entities, since popular entities may receive overall more content from external description sources than tail entities.

Dynamicity. Dynamic external description sources cause the entity’s descriptions to change in near real-time. Consequently, a static ranking model cannot capture the evolving and continually changing index that follows from our dynamic scenario, hence we employ an adaptive ranking model that is continuously updated.

5.3 Model

In the following section, we describe our supervised ranking approach, by first explaining the entity representation, the set of features we employ for learning an optimal representation for retrieval, and finally the supervised method.

5.3.1 Entity Representation

To deal with dynamic entity representations, which are composed of content from different external description sources, we model entities as fielded documents:

$$e = \{\bar{f}_{title}^e, \bar{f}_{text}^e, \bar{f}_{anchors}^e, \dots, \bar{f}_{query}^e\}. \quad (5.1)$$

Here, \bar{f}^e corresponds to the field term vector that represents e ’s content from a single source (denoted in subscript). We refer to this collection of field term vectors as the entity’s *representation*.

The fields with content from dynamic description sources may change over time. We refer to the process of adding an external description source’s term vector to the entity’s

corresponding field term vector as an *update*. To model these dynamically changing fields, we discretize time (i.e., $T = \{t_1, t_2, t_3, \dots, t_n\}$), and define *updating* fields as:

$$\bar{f}_{query}^e(t_i) = \bar{f}_{query}^e(t_{i-1}) + \begin{cases} \bar{q}, & \text{if } e_{clicked} \\ 0, & \text{otherwise} \end{cases} \quad (5.2)$$

$$\bar{f}_{tweets}^e(t_i) = \bar{f}_{tweets}^e(t_{i-1}) + \overline{tweet_e} \quad (5.3)$$

$$\bar{f}_{tags}^e(t_i) = \bar{f}_{tags}^e(t_{i-1}) + \overline{tag_e}. \quad (5.4)$$

In equation 5.2, \bar{q} represents the term vector of query q that is summed element-wise to e 's query field term vector (\bar{f}_{query}^e) at time t_i , if e is clicked by a user that issues q . In equation 5.3, $\overline{tweet_e}$ represents the field term vector of a tweet that contains a link to the Wikipedia page of e , which also gets added element-wise to the corresponding field (\bar{f}_{tweets}^e). Finally, in equation 5.4, $\overline{tag_e}$ is the term vector of a tag that a user assigns to the Wikipedia page of e .

To estimate e 's relevance to a query q given the above-described representation one could, e.g., linearly combine retrieval scores between q 's term vector \bar{q} and each $\bar{f} \in e$. However, due to the heterogeneity that exists both between the different fields that make up the entity representation, and between different entities (described in Section 5.2.4), linearly combining similarity scores may be sub-optimal [223], and hence we employ a supervised single-field weighting model [162]. Here, each field's contribution towards the final score is individually weighted, through learned field weights from implicit user feedback.

5.3.2 Features

To learn the optimal entity representation for retrieval, we employ three types of feature that express field and entity importance: first, *field similarity* features are computed per field and boil down to query–field similarity scores. Next, *field importance* features, likewise computed per field, aim to inform the ranker of the status of the field at that point in time (i.e., to favor fields with more and novel content). Finally, we employ *entity importance* features, which operate at the entity level and aim to favor recently updated entities.

Field Similarity

The first set of features model the similarity between a query and a field, which we denote as ϕ_{sim} . For query–field similarity, we compute TF×IDF cosine similarity. We define

$$\phi_{sim}(q, \bar{f}, t_i) = \sum_{w \in q} n(w, \bar{f}(t_i)) \cdot \log \frac{|C^{t_i}|}{|\{\bar{f}(t_i) \in C^{t_i} : w \in \bar{f}(t_i)\}|}, \quad (5.5)$$

where w corresponds to a query term, $n(w, \bar{f}(t_i))$ is the frequency of term w in field \bar{f} at time t_i . C^{t_i} is the collection of fields at time t_i , and $|\cdot|$ indicates set cardinality. More elaborate similarity functions can be used, e.g., BM25(F), however, we choose a parameter-less similarity function that requires no tuning. This allows us to directly compare the contribution of the different expansion fields without having additional

factors play a role, such as length normalization parameters, which affect different fields in non-trivial ways.

Field Importance

The next set of features is also computed per field, ϕ_p is meant to capture a field's *importance* at time t_i :

$$\phi_p(\bar{f}(t_i), e) = S(\bar{f}(t_i), e). \quad (5.6)$$

We instantiate four different field importance features. First, we consider two ways to compute a field's length, either in terms (5.7) or in characters (5.8):

$$\phi_{p_1}(\bar{f}(t_i), e) = |\bar{f}(t_i)| \quad (5.7)$$

$$\phi_{p_2}(\bar{f}(t_i), e) = \sum_{w \in \bar{f}(t_i)} |w| \quad (5.8)$$

The third field importance scoring function captures a field's novelty at time t_i , to favor fields that have been updated with previously unseen, newly associated terms to the entity (i.e., terms that were not in the original entity representation at t_0):

$$\phi_{p_3}(\bar{f}(t_i), e) = |\{w \in \bar{f}(t_i) : w \notin \bar{f}(t_0)\}|. \quad (5.9)$$

The fourth field importance scoring function expresses whether a field has undergone an update at time t_i :

$$\phi_{p_4}(\bar{f}, t_i, e) = \sum_{j=0}^i + = \begin{cases} 1, & \text{if } \text{update}(\bar{f}(t_j)) \\ 0, & \text{otherwise,} \end{cases} \quad (5.10)$$

where $\text{update}(\bar{f}(t_j))$ is a Boolean function indicating whether field f was updated at time j , i.e., we sum from t_0 through t_i and accumulate updates to the fields.

Entity Importance

The feature ϕ_I models the entity's importance. We compute the time since the entity last received an update to favor recently updated entities:

$$\phi_I(e, t_i) = t_i - \max_{\bar{f} \in e} \text{time}_{\bar{f}}, \quad (5.11)$$

here, t_i is the current time, and $\text{time}_{\bar{f}}$ is the update time of field \bar{f} . $\max_{\bar{f}}$ corresponds to the timestamp of the entity's field that was most recently updated.

5.3.3 Machine Learning

Given the features explained in the previous section, we employ a supervised ranker to learn the optimal feature weights for retrieval:

$$\Omega = (\omega_e, \omega_{\bar{f}_{title}}, \omega_{p1_{title}}, \omega_{p2_{title}}, \dots, \omega_{\bar{f}_{text}}, \omega_{p1_{text}}, \dots). \quad (5.12)$$

Here, Ω corresponds to the weight vector, which is composed of individual weights (ω) for each of the field features (similarity and importance) and the entity importance feature.

To find the optimal Ω , we train a classification-based re-ranking model, and learn from user interactions (i.e., clicks). The model employs the features detailed in the previous section, and the classification’s confidence score is used as a ranking signal. As input, the ranker receives a feature vector (x), extracted for each entity-query pair. The associated label y is positive (1) for the entities that were clicked by a user who issued query q . We define x as

$$x = \{\phi_{sim_1}, \phi_{p1_1}, \phi_{p2_1}, \phi_{p3_1}, \phi_{p4_1}, \dots, \phi_{sim_{|e|}}, \phi_{p1_{|e|}}, \phi_{p2_{|e|}}, \phi_{p3_{|e|}}, \phi_{p4_{|e|}}, \phi_I\}, \quad (5.13)$$

where $|e|$ corresponds to the number of fields that make up the entity representation ($\bar{f} \in e$).

See Algorithm 1 for an overview of our machine learning method in pseudo-code. As input, our supervised classifier is given a set of $\langle q, e, L \rangle$ -tuples (see l. 4). These tuples consist of a query (q), a candidate entity (e), and a (binary) label (L): positive (1) or negative (0). Given an incoming query, we first perform top- k retrieval (see Section 5.4.4 for details) to yield our initial set of candidate entities: $\mathcal{E}_{candidate}$ (l. 8). For each entity, we extract the features which are detailed in Section 5.3.2 (l. 12) and have the classification-based ranker R output a confidence score for e belonging to the positive class, which is used to rank the candidate entities (l. 13). Labels are acquired through user interactions, i.e., entities that are in the set of candidate entities and clicked after issuing a query are labeled as positive instances (l. 16–22), used to retrain R (l. 24). Finally, after each query, we allow entities to be updated by dynamic description sources: tweets and tags (l. 26), we provide more details in Section 5.4.3.

5.4 Experimental setup

In this section, we start by describing our experiments and how they allow us to answer the three subquestions raised in Section 5.1, and then, we present our baselines, machine learning setting, and describe our evaluation methodology.

5.4.1 Experiments

Experiment 1. To answer our first subquestion (RQ3.1): *does entity ranking effectiveness increase using dynamic collective entity representations?* we compare our proposed dynamic collective entity ranking method to a baseline that only incorporates KB fields for entity ranking: KBER (Knowledge Base Entity Representations). We restrict both the baseline and our Dynamic Collective Entity Representation (DCER) method to the set of field similarity features (Section 5.3.2), which we denote as $KBER_{sim}$ and $DCER_{sim}$. This allows us to provide clear insights into the contribution of the fields’ content in ranking effectiveness. In addition, we perform an ablation study and compare the similarity-baseline to several approaches that incorporate content from a single external description source (denoted $KB+source_{sim}$).

Algorithm 1 Pseudo-algorithm for learning optimal Ω^T .

Require: Ranker R , Knowledge Base KB , Entities \mathcal{E}

```

1:  $\mathcal{E} \leftarrow \{e_1, e_2, \dots, e_{|KB|}\}$ 
2:  $e \leftarrow \{\bar{f}_{title}, \bar{f}_{anchors}, \dots, \bar{f}_{text}\}$ 
3:
4:  $\mathcal{L} = \{\langle q_1, e_1, \{0, 1\} \rangle, \langle q_1, e_2, \{0, 1\} \rangle, \dots, \langle q_n, e_m, \{0, 1\} \rangle\}$ 
5:  $R \leftarrow \text{Train}(\mathcal{L})$ 
6:
7: while  $q$  do
8:    $\mathcal{E}_{candidate} \leftarrow \text{Top-}k \text{ retrieval}(q)$ 
9:    $\mathcal{E}_{ranked} \leftarrow []$ 
10:
11:   for  $e \in \mathcal{E}_{candidate}$  do
12:      $\phi_e \leftarrow \text{Extract features}(e)$ 
13:      $\mathcal{E}_{ranked} \leftarrow \text{Classify}(R, \phi_e)$ 
14:   end for
15:
16:    $e_{clicked} \leftarrow \text{Observe click}(q, \mathcal{E})$ 
17:   if  $e_{clicked} \in \mathcal{E}_{candidate}$  then
18:      $\mathcal{L} \leftarrow \mathcal{L} \cup \{\langle q, e_{clicked}, 1 \rangle\}$ 
19:      $e_{clicked} \leftarrow e_{clicked} \cup \{text_q\}$ 
20:   else
21:      $\mathcal{L} \leftarrow \mathcal{L} \cup \{\langle q, e_{clicked}, 0 \rangle\}$ 
22:   end if
23:
24:    $R \leftarrow \text{Train}(\mathcal{L})$ 
25:   for  $e \in \mathcal{E}$  do
26:      $e \leftarrow e \cup \{\bar{f}_{e,tweet_1}, \bar{f}_{e,tag_1}, \dots, \bar{f}_{e,tweet_i}, \bar{f}_{e,tag_j}\}$ 
27:   end for
28: end while

```

Experiment 2. We address RQ3.2, *does entity ranking effectiveness increase when employing field and entity features*, by comparing the KBER_{sim} baseline that only incorporates field similarity features, to the KBER baseline that incorporates the entity and field importance features, and to our DCER method.

Experiment 3. Finally, to answer RQ3.3, *does entity ranking effectiveness increase when we continuously learn the optimal entity representations?* we compare our proposed entity ranking method DCER to its non-adaptive counterpart (DCER_{na}), that we do not periodically retrain. Contrasting the performance of this non-adaptive system with our adaptive system allows us to tease apart the effects of adding more training data, and the effect of the additional content that comes from dynamic external description sources. Here too, we include an ablation study, and compare to non-adaptive approaches that incorporate content from a single external description source (denoted $\text{KB}+\text{source}_{na}$).

5.4.2 Baselines

Due to our focus on dynamic entity representations and adaptive rankers, running our method on datasets from seemingly related evaluation campaigns such as those in TREC [13, 14] and INEX [57–59] is not feasible. We are constrained by the size of datasets, i.e., we need datasets that are sufficiently large (thousands of queries) and time-stamped, which excludes the aforementioned evaluation campaigns, and hence direct comparison to results obtained there. In a scenario with dynamic fields and continually changing content, our method is required to reweigh fields continuously to reflect changes in the fields’ content. Continually (re-)tuning parameters of existing fielded retrieval methods such as BM25F [223], when documents in the index change is exceedingly expensive, rendering these methods unsuitable for this scenario.

For these reasons we consider the following supervised baselines in our experiments; KBER_{sim} is an online learning classification-based entity ranker, that employs field similarity features on entity representations composed of KB description sources (i.e., title, text, categories, anchors, redirects and links fields). KBER is the same baseline system, extended with the full set of features (described in Section 5.3.2). Finally, DCER_{na} is a non-adaptive baseline: it incorporates all external description sources, and all features as our proposed DCER method, but does not periodically retrain.

5.4.3 Data Alignment

In our experiments we update the fields that jointly represent an entity with external descriptions that come in a streaming manner, from a range of heterogeneous external description sources. This assumes that all data sources run in parallel in terms of time. In a real-world setting this assumption may not always hold as systems need to integrate historical data sources that span different time periods and are of different size for bootstrapping. We simulate this scenario by choosing data sources that do not originate from the same time period nor span the same amount of time (see also Table 5.1). To remedy this, we introduce a method for time-aligning all data sources (which range from 2009 to 2014) to the timeline of the query log (which dates from 2006). We apply a *source-time* transformation to mitigate the dependence of content popularity on time and when it was created [240, 244]. Each query is treated as a time unit, and we distribute the expansions from the different sources over the queries, as opposed to obeying the misaligned timestamps from the query log and expansion sources.

To illustrate: given n queries and a total of k items for a given expansion source, after each query we update the entity representations with $\frac{n}{k}$ expansions of that particular expansion source. In our dataset we have 32,901 queries, 52,631 tweets, and 4,429,692 tags. After each query we distribute 1 query, 2 tweets, and 135 tags over the entities in the index. Mapping real time to source time evenly spreads the content of each data source within the timespan of the query log. This smooths out bursts but retains the same distribution of content over entities (in terms of, e.g., entity popularity). The diminishing effect on burstiness is desirable in the case of misaligned corpora, as bursts are informative of news events, which would not co-occur in the different data sources. Although our re-aligned corpora cannot match the quality of real parallel corpora, our

method offers a robust lower bound to understand the utility of collective dynamic entity representations of our method. We reiterate that our goal in this chapter is to study the effect of dynamic entity representations and adapting rankers, not to leverage temporal features for improving entity retrieval.

5.4.4 Machine Learning

We apply machine learning for learning to weight the different fields that make up an entity representation for optimal retrieval effectiveness. In response to a query, we first generate an initial set of candidate entities by retrieving the top- k entities to limit the required computational resources for extracting all features for all documents in the collection [159]. Our top- k retrieval method involves ranking all entities using the similarity function described in Section 5.3.2, where we collapse the fielded entity representation into a single document. We set $k = 20$ as it has shown a fair tradeoff between high recall (80.1% on our development set) and low computational expense. We choose Random Forests as our machine learning algorithm because it has proven robust in a range of diverse tasks (e.g., [187]), and can produce confidence scores that we employ as ranking signal. In our experiments, we set the number of trees to 500 and the number of features each decision tree considers for the best split to $\sqrt{|\Omega|}$.

5.4.5 Evaluation

For evaluating our method’s adaptivity and performance over time, we create a set of incremental time-based train/test splits as in [22]. We first split the query log into K chunks of size N : $\{C_1, C_2, C_3, \dots, C_K\}$. We then allocate the first chunk (C_1) for training the classifier and start iteratively evaluating each succeeding query. Once the second chunk of queries (C_2) has been evaluated, we expand the training set with it and retrain the classifier. We then continue evaluating the next chunk of queries (C_3). This procedure is repeated, continually expanding the training set and retraining the classifier with N queries (we set $N=500$ in our experiments). In this scenario, users’ clicks are treated as ground truth and the classifier’s goal is to rank clicked entities at position 1. We do not distinguish between clicks (e.g., satisfied clicks and non-satisfied clicks), and we leave more advanced user models, e.g., which incorporate skips, as future work.

To show the robustness of our method we apply five-fold cross-validation over each run, i.e., we generate five alternatively ordered query logs by shuffling the queries. We keep the order of the dynamic description sources fixed to avoid conflating the effect of queries’ order with that of the description sources.

Since we are interested in how our method behaves over time, we plot the MAP at each query over the five-folds, as opposed to averaging the scores over all chunks across folds (as in [22]) and losing this temporal dimension. In addition to reporting MAP, we report on P@1, as there is only a single relevant entity per query in our experimental setup. We test for statistical significance using a two-tailed paired t-test. Significant differences are marked \blacktriangle for $\alpha = 0.01$.

Table 5.2: Performance of field similarity-based entity ranking methods using KB entity representations (KBER_{sim}) and dynamic collective entity representations (DCER_{sim}) in terms of MAP and P@1. Significance tested against KBER_{sim} . Oracle marks upper bound performance given our top- k scenario.

Run	MAP	P@1
KBER_{sim}	0.5579	0.4967
DCER_{sim}	0.5971[▲]	0.5573[▲]
Oracle	0.6653	0.6653

5.5 Results and Analysis

We report on the experimental results for each of our three experiments in turn, and provide an analysis of the results to better understand the behavior of our method.

5.5.1 Dynamic Collective Entity Representations

In our first experiment, we explore the impact of the description sources we use for constructing dynamic collective entity representations, and aim to answer RQ3.1. We compare the KBER_{sim} baseline, which incorporates field similarity on KB descriptions, to our DCER_{sim} method, which incorporates field similarity features on all description sources (web anchors, tweets, tags, and queries). Table 5.2 shows the performance in terms of MAP and P@1 of the baseline (KBER_{sim}) and DCER_{sim} after observing all queries in our dataset. We include an oracle run as a performance upper bound given our top- k retrieval scenario.

The results show that the dynamic collective entity representations manage to significantly outperform the KB entity representations for both metrics, and that DCER_{sim} presents the correct entity at the top of the ranking for over 55% of the queries.

Next, we look into the impact on performance of individual description sources for our dynamic collective entity representations. We add each source individually to the KBER_{sim} baseline. Figure 5.3 shows how each individual description source contributes to produce a more effective ranking, with KB+tags narrowly outperforming KB+web as the best single source. Combining all sources into one, yields the best results, outperforming KB+tags by more than 3%. We observe that after about 18,000 queries, KB+tags overtakes the (static) KB+web method, suggesting that newly incoming tags yield higher ranking effectiveness. All runs show an upward trend and seem to level out around the 30,000th query. This pattern is seen across ranking methods, which indicates that the upward trend can be attributed to the addition of more training data (queries).

Table 5.3 lists the results of all methods along with the improvement rate (relative improvement when going from 10,000 to all queries). The runs that incorporate dynamic description sources (i.e., KB+tags, KB+tweets, and KB+queries) show the highest relative improvements (at 6.4%, 6.1% and 6.9% respectively). Interestingly, for P@1, the KB+queries method yields a substantial relative improvement (+8.9%), indicating that the added queries provide a strong signal for ranking the clicked entities at the top.

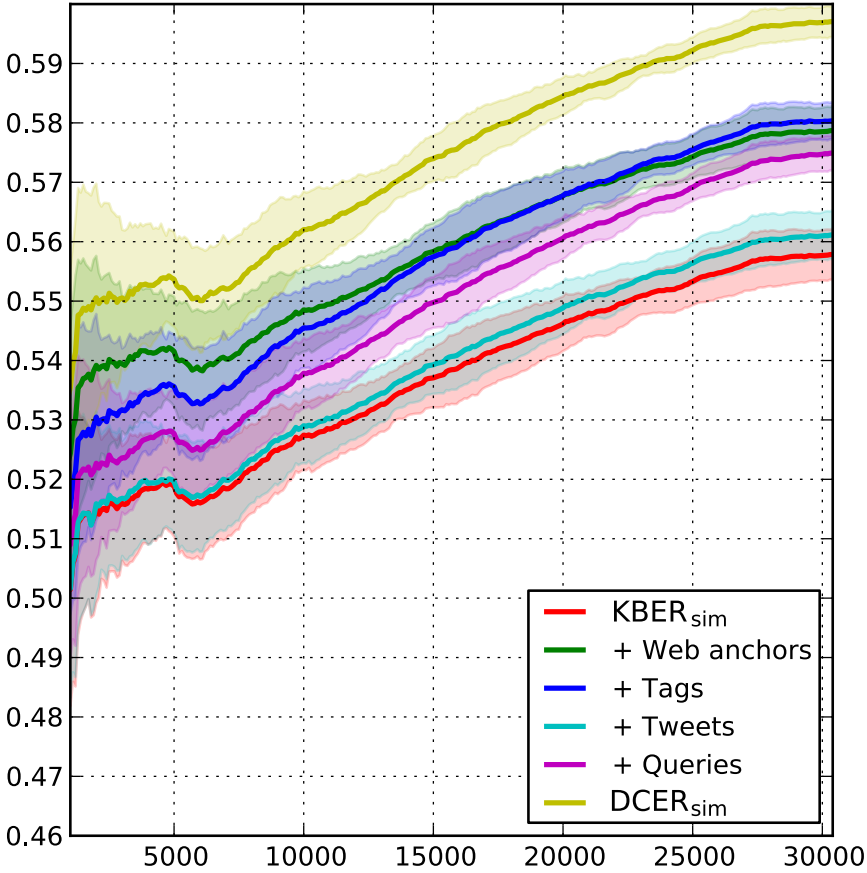


Figure 5.3: Impact on performance of individual description sources. MAP on the y-axis, number of queries on the x-axis. The line represents MAP, averaged over 5-folds. Standard deviation is shown as a shade around the line. This plot is best viewed in color.

The comparatively lower learning rates of methods that incorporate only static description sources (KBER_{sim} and KB+web yield relative improvements of +5.8% and +5.5%, respectively), suggest that the entity content from dynamic description sources effectively contributes to higher ranking performance as the entity representations change and the ranker is able to reach a new optimum. DCER_{sim}, the method that incorporates all available description sources, shows a comparatively lower relative improvement, which is likely due to it hitting a ceiling, and not much relative improvement can be gained.

Feature Weights over Time

A unique property in our scenario is that, over time, more training data is added to the system, and more descriptions from dynamic sources come in, both of which are expected

Table 5.3: Comparison of relative improvement between runs with different field similarity features. We report on MAP and P@1 at query 10,000 and the last query. Rate corresponds to the percentage of improvement between the 10,000th and final query. Significance tested against KBER_{sim} .

Run	MAP (10k)	MAP (end)	Rate	P@1 (10k)	P@1 (end)	Rate
KBER_{sim}	0.5274	0.5579	+5.8%	0.4648	0.4967	+6.9%
KB+Web _{sim}	0.5485	0.5787 [▲]	+5.5%	0.4965	0.5282 [▲]	+6.4%
KB+Tags _{sim}	0.5455	0.5804 [▲]	+6.4%	0.4930	0.5317 [▲]	+7.8%
KB+Tweets _{sim}	0.5290	0.5612 [▲]	+6.1%	0.4673	0.5021 [▲]	+7.5%
KB+Queries _{sim}	0.5379	0.5750 [▲]	+6.9%	0.4813	0.5242 [▲]	+8.9%
DCER _{sim}	0.5620	0.5971[▲]	+6.2%	0.5178	0.5573[▲]	+7.6%

to improve the performance of our method. To better understand our the behavior of our method, we look into the learned feature weights for both a static approach and our dynamic one at each retraining interval. We consider as feature importance the average height of a feature when it is used as a split-node in one of the trees that make up the random forest. The underlying intuition is that features higher up a decision tree affect a larger fraction of the samples, and hence can be considered more important. Figure 5.4 shows the weights for six static fields (categories, title, anchors, links, text, redirects) and three dynamic fields (queries, tweets, tags).

The KBER_{sim} baseline shows little change in the feature weights as more training data is added. The anchors’ weight increases slightly, while the text and categories fields’ weights steadily decline over time. The latter two features show a steadily decreasing standard deviation, indicating that the ranker becomes more confident in the assigned weights.

For KB+queries it is apparent that the queries field weight increases substantially as both the ranking method receives more training data and the entity representations receive more content from the queries description source. At the same time, anchors, redirects, and title field weights, which were assigned consistently high weights in KBER_{sim} , seem to pay for the increase of the importance of queries. Text, category, and links show similar patterns to the baseline run; steadily declining weights, converging in terms of a decreasing standard deviation.

The KB+tags run shows a similar pattern as KB+queries: we observe an increase over time of the weight assigned to the field that holds content from the tag description source, at the cost of original KB fields, resulting in improved ranking effectiveness. Looking at KB+tweets, however, we observe a different pattern. The tweets field starts out with a very low weight, and although the weight steadily increases, it remains low. The ranker here, too, becomes more confident on this source’s weight, with the standard deviation dissolving over time. Looking at the higher performance of KB+tweets in comparison to the KBER_{sim} baseline, together with the field’s weight increasing over time, we conclude that the tweets added to the entity representations provide added value.

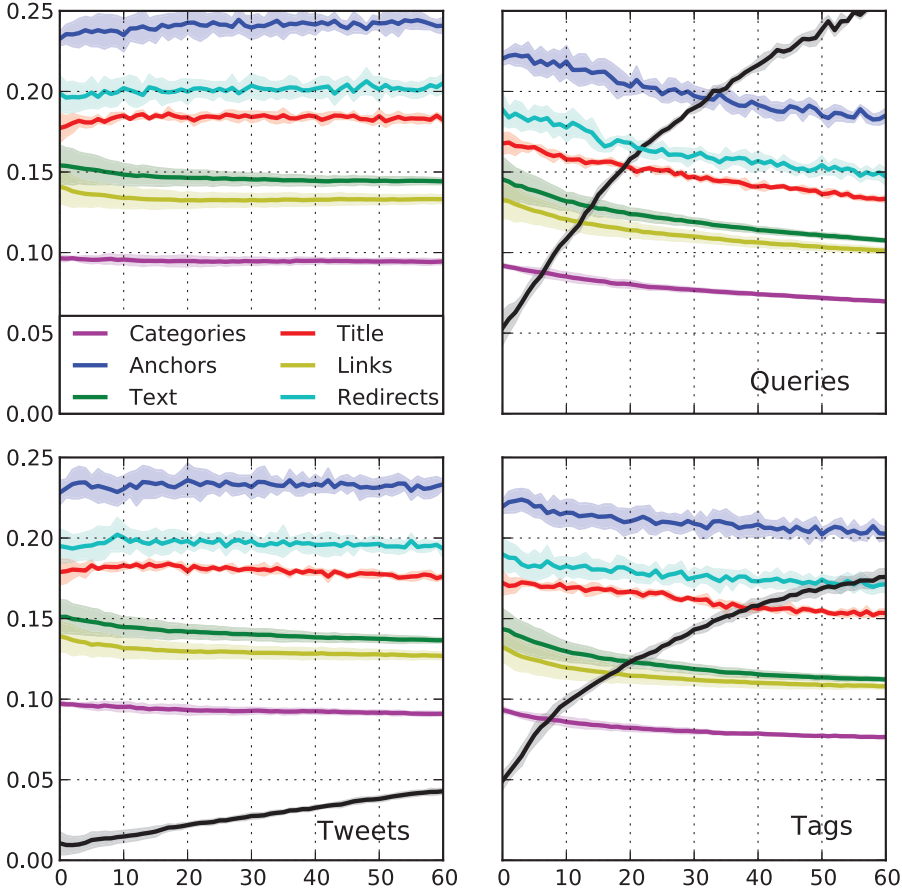


Figure 5.4: Feature weights over time: y-axis shows (relative) feature weights, x-axis shows each chunk of 500 queries where the ranker is retrained. Starting from top left in a clock-wise direction we show the following runs: KBER_{sim} (baseline), KB+Queries, KB+Tags, KB+Tweets. The black line shows the dynamic description source’s weight. This plot is best viewed in color.

5.5.2 Modeling Field Importance

In our second experiment, which aims to answer RQ3.2, we turn to the contribution of the additional field and entity importance features. Table 5.4 lists the results of the best performing run with only field similarity features (DCER_{sim}), the KBER baseline that incorporates the full set of features, and our proposed DCER method which likewise incorporates the full set of features.

The results show that modeling field and entity importance significantly improves effectiveness of both the KBER_{sim} baseline and the DCER_{sim} runs. After running through

Table 5.4: Comparison of relative improvement between the KBER baseline with field importance features for KB fields, and our DCER method with these features for all fields. We also show the best performing run without field and entity importance features. Significance tested against KBER.

Run	MAP (10k)	MAP (end)	Rate	P@1 (10k)	P@1 (end)	Rate
Oracle	–	0.6653	–	–	0.6653	–
<small>DCER_{sim}</small>	0.5620	0.5971	+6.2%	0.5178	0.5573	+7.6%
<small>KBER</small>	0.5853	0.6129	+4.7%	0.5559	0.5831	+4.9%
<small>DCER</small>	0.5923	0.6200[▲]	+4.7%	0.5655	0.5925[▲]	+4.8%

the entire dataset, the performance of DCER approaches that of the oracle run which also explains why the differences between the two approaches here is rather small: we are very close to the maximum achievable score.

Figure 5.5 shows the performance of the two approaches over time. The pattern is similar to the one in Section 5.5.1: both lines show a steady increase, while DCER maintains to be the best performing.

5.5.3 Ranker Adaptivity

In our third experiment, where we aim to answer RQ3.3, we compare our adaptive ranker (DCER), which is continuously retrained, to a non-adaptive baseline (DCER_{na}), which is only trained once at the start and is not retrained.

Table 5.5: Comparing relative improvement between runs with and without an adaptive ranker. Statistical significance tested between the KBER and DCER approaches and their non-adaptive counterparts.

Run	Adaptive	MAP	P@1
<small>KBER_{na}</small>	no	0.5198	0.4392
<small>KBER</small>	yes	0.5579 [▲]	0.4967 [▲]
<small>DCER_{na}</small>	no	0.5872	0.5408
<small>DCER</small>	yes	0.5971[▲]	0.5573[▲]

Results in Table 5.5 show that incrementally re-training the ranker is beneficial to entity ranking effectiveness. For both KBER and DCER we see a substantial improvement when moving from one single batch training to continuously retraining the ranker.

To better understand this behavior, we plot in Figure 5.6 the performance of all runs we consider over time and at the same time; Table 5.6 provides the detailed scores. Broadly speaking we observe similar patterns between adaptive and non-adaptive methods, and we identify three interesting points. First, for the non-adaptive methods, the absolute performance is substantially lower across the board. Second, for the adaptive methods,

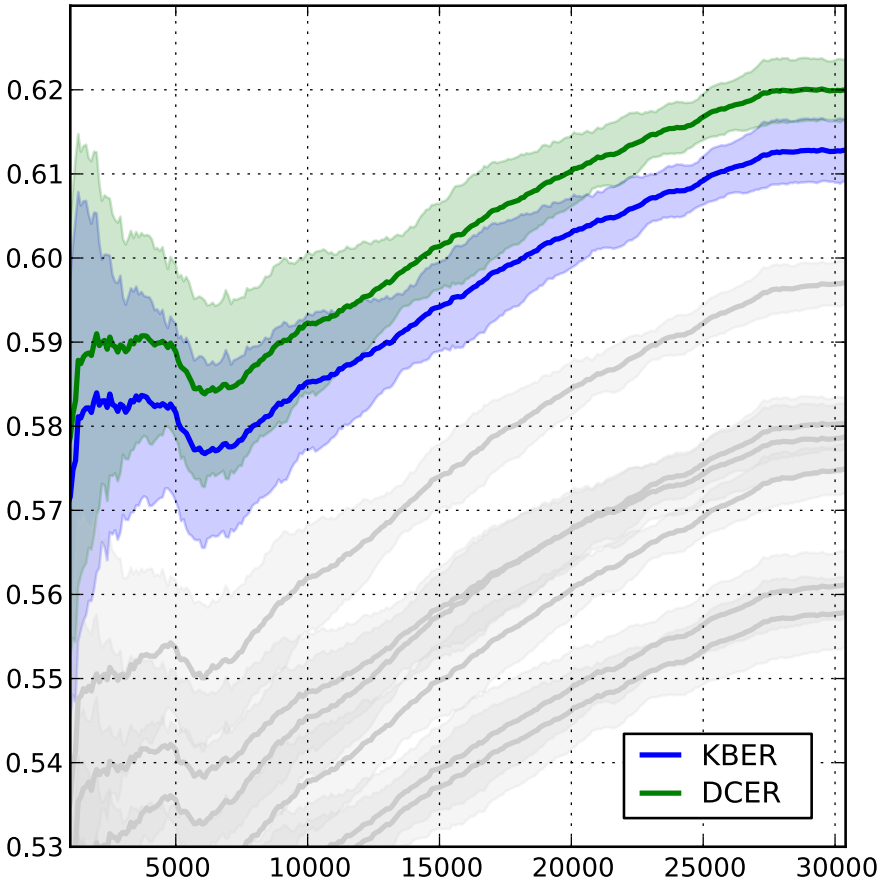


Figure 5.5: Runs with field similarity, field importance, and entity importance features. MAP on the y-axis, number of queries on the x-axis. The line represents MAP, averaged over 5-folds. Standard deviation is shown as a shade around the line. This plot is best viewed in color.

the standard deviation (as shown in Figures 5.3 and 5.6) is substantially lower, which indicates that retraining increases the ranking method’s confidence in optimally combining the different descriptions for the entity representation. Third, the learning rates in the adaptive setting are substantially higher, reaffirming our observation that learning a single global feature weight vector is not optimal.

Table 5.6 shows that the difference in learning rates between methods that only incorporate static description sources (KBER, KB+web) and methods that incorporate dynamic sources is pronounced, in particular for the tags and queries sources. This indicates that even with fixed feature weights, the content that comes in from the dynamic description sources yields an improvement in entity ranking effectiveness. Finally, the

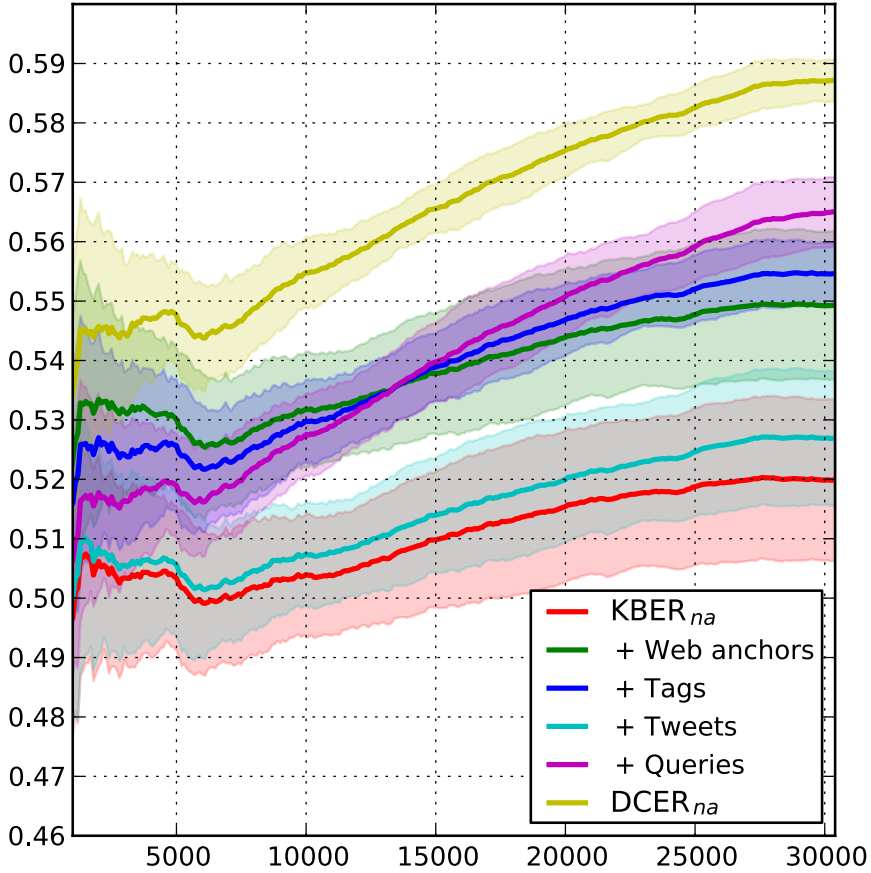


Figure 5.6: Individual description sources with non-adaptive ranker. MAP on the y-axis, number of queries on the x-axis. The line represents MAP, averaged over 5-folds. Standard deviation is shown as a shade around the line. This plot is best viewed in color.

methods that only incorporate static description sources also show lower learning rates than their adaptive counterparts, which indicates that retraining and adapting feature weights is desirable even with static entity representations.

5.6 Conclusion

In this chapter, we addressed the entity ranking task, and set out to improve it by enriching entity representations from entity descriptions collected from a wide variety of sources. We set out to answer the following question:

5. Retrieving Emerging Entities

Table 5.6: Comparing relative improvement between non-adaptive runs. Significance tested against KBER_{na} .

Run	MAP (10k)	MAP (end)	Rate	P@1 (10k)	P@1 (end)	Rate
KBER_{na}	0.5040	0.5198	+3.1%	0.4286	0.4392	+2.5%
KB+Web_{na}	0.5318	0.5493 [▲]	+3.3%	0.4698	0.4829 [▲]	+2.8%
KB+Tags_{na}	0.5298	0.5546 [▲]	+4.7%	0.4671	0.4904 [▲]	+5.0%
KB+Tweets_{na}	0.5074	0.5269 [▲]	+3.8%	0.4334	0.4490 [▲]	+3.6%
KB+Queries_{na}	0.5275	0.5650 [▲]	+7.1%	0.4659	0.5090 [▲]	+9.2%
DCER_{na}	0.5548	0.5872[▲]	+5.8%	0.5063	0.5408 [▲]	+6.8%

RQ3 Can we leverage collective intelligence to construct entity representations for increased retrieval effectiveness of entities of interest?

We have shown an effective way of leveraging collective intelligence to construct entity representations for increased retrieval effectiveness of entities of interest.

To answer RQ3 we formulated and answered (in Section 5.5) three subquestions. First, we aim to answer:

RQ3.1 Does entity ranking effectiveness increase by using dynamic collective entity representations?

We demonstrate that incorporating dynamic description sources into dynamic collective entity representations enables a better matching of users’ queries to entities, resulting in an increase of entity ranking effectiveness. More specifically, we show how each individual description source contributes to produce a more effective ranking (see Figure 5.3). Looking at the contribution of individual description sources, we see how the highest relative improvements come from dynamic description sources, and social tags prove to be the strongest individual signal. In addition, we study how the ranker adapts the individual weights of fields (reflecting the relative importances of expansion sources) in Section 5.5.1, and see how dynamic description sources gain higher importances as new descriptions come in.

Next, we answer subquestion RQ3.2:

RQ3.2 Does entity ranking effectiveness increase when employing additional field and entity importance features?

We show that informing the ranker on the expansion state of the entities, i.e., employing field and entity importance features, further increases the ranking effectiveness.

Finally, we answer:

RQ3.3 Does entity ranking effectiveness increase when we continuously learn the optimal way to combine the content from different description sources?

We compare an adaptive ranker (i.e., one that we periodically retrain, as descriptions come in) to a static one. The results show how retraining the ranker leads to improved ranking effectiveness in dynamic collective entity representations. We also show how even the static ranker improves over time, suggesting that even the static ranker benefits from newly incoming descriptions, if not as much as the adaptive method.

The findings in this work have several implications in the discovery context. Namely, we have shown that we can effectively leverage the feedback of searchers (i.e., their queries and clicks) to improve retrieval effectiveness, which may be beneficial in the discovery process. Furthermore, we have shown that we can effectively combine heterogeneous information related to an entity of interest into a single representation, which suggests our method can be useful in a scenario where data is available from many different sources, a scenario not uncommon in the context of E-Discovery. Consider, e.g., the scenario where emails and attachments in various different formats needs to be considered, a scenario that is not uncommon in E-Discovery [197]. By combining the content that is explicitly linked (e.g., attachments to email addresses), we have shown that regardless of the nature of the data, our ranking method can effectively incorporate the additional data for improved retrieval effectiveness.

The work presented in this chapter is not without limitations, however. Our study of the impact of dynamic collective entity representations was performed in a controlled scenario, where we collect and leverage different data sources. One restriction of working with these freely available resources is that it proves hard to find aligned and sizeable datasets. In particular, the temporal misalignment between different corpora prevents the analysis of temporal patterns that may span across sources (e.g., queries and tweets showing similar activity around entities when news events unfold). In part, these restrictions can be circumvented in future work, e.g., increasing the (comparatively) low number of tweets by enriching them through e.g., entity linking methods [176], where entities identified in tweets could be expanded. Additional challenges and opportunities may arise when increasing the scale of the data collections. Opportunities may lie in exploiting session or user information for more effective use of user interaction signals. Challenges include so-called “swamping” [223] or “document vector saturation” [138], i.e., entity drift that is more prone to happen when the size of the data collections increase.

A natural extension to this work is to study this problem in a realistic scenario, with aligned datasets of sufficient size, which would allow us to study the impact of “swamping” entities in more detail, as well as provide a more fine-grained analysis of the impact of the importance features. But more importantly, it would allow to study the link between the temporal patterns of entities (as studied in Chapter 3) and their retrieval effectiveness. Novel features that take into account the novelty in fields (e.g., the number of terms that were not in the original entity description), and the diversity could prove useful in adapting for entity drift. Furthermore, whilst we consider the changing entity representation now by its impact on retrieval effectiveness, another follow-up may investigate the changes in entity representations on a qualitative level, e.g., by studying per field and/or entity the type of additional descriptions (similar to original vs. novel), the rate of changing representations, etc.

Part II

Analyzing and Predicting Activity from Digital Traces

6

Analyzing and Predicting Email Communication

“I speak to everyone in the same way,
whether he is the garbage man or the president of the university.”

—Albert Einstein

6.1 Introduction

In this first chapter of this part of the thesis, we present our first case study into predicting activities of entities of interest from digital traces. More specifically, we propose a method for predicting the recipient of an email, by leveraging both the contexts in which the digital traces are created, as their content.

Knowing the likely recipients of an email can be helpful from a user-perspective (e.g., for pro-actively suggesting recipient), but also in the context of discovery. Reliably predicting communication flows may find applications in discovering *entities of interest*, e.g., by comparing actual communication flows to “expected” (i.e., predicted) flows, one can identify communication that is “unexpected,” which can be valuable signal in the context of digital forensics [200].

Despite the huge increase in the use of social media channels and online collaboration platforms, email remains one of the most popular ways of (online) communication. Email traffic has been increasing steadily over the past few years, and recent market studies have projected its continued growth for the years to come [212]. Recipient recommendation methods aim at providing the sender of an email with the appropriate predicted recipients of the email that is currently being written. These methods furthermore allow us to gain a better understanding of communication patterns in enterprises, potentially revealing underlying structures of communication networks.

In this chapter, we focus on recipient prediction in an enterprise setting, as it allows us to leverage the full content and structure of the communication network, as opposed to strictly local (ego network) approaches, that are restricted to only the sender’s direct network. Furthermore, enterprise email databases are common objects of study in the E-Discovery scenario, as can be seen by the TREC Legal track which aims to closely

resemble a real-world E-Discovery task [46, 117]. Henseler [118] likewise studied email in an E-Discovery setting. He found that combining keyword search with communication pattern analysis reduces the amount of information reviewers need to process.

We propose a novel hybrid generative model, which incorporates both the communication graph structure and email content signals for recipient prediction. Our model predicts recipients from scratch, i.e., without assuming seed recipients for prediction. Furthermore, our model can quickly deal with updates in both the communication graph and the profiles of recipients when new emails are sent around.

Our main research question revolves around the contribution of *context* of digital sources (i.e., social context or organizational structure) and *content* of digital sources (i.e., the actual email content) towards effectiveness of predicting communication. More specifically, this chapter aims to answer RQ4:

RQ4 Can we predict email communication through modeling email content and communication graph properties?

On top of that we investigate how we can optimally estimate the various components of our model. We train our recipient prediction model on one email collection (the Enron email collection) and test it on another (the Avocado email collection).

Our main finding is that a combination of the communication graph and email content outperforms the individual components. We obtain optimal performance when we incorporate the number of emails a recipient has received so far and the number of emails a given sender sent to a recipient at that point in time in our model. Other options, like using PageRank as a recipient prior, do not improve performance. Our findings show that both the *context* and *content* of digital sources contribute to prediction effectiveness.

6.2 Communication Graph

To model the *context* in which communication happens, we construct a communication graph from all emails sent by users in our email collections. This graph implicitly captures the (social or professional) relations between users in an email network, and hence models an aspect of the context of communication.

We consider the email traffic as a directed graph G , consisting of a set of vertices and arcs (directed edges) $\langle V, A \rangle$. Each vertex $v \in V$ represents a distinct email address in the corpus (i.e., a sender S or recipient R in terms of our modeling in Section 6.3), and arcs $a \in A$ that connect them represent the communication between the two corresponding addresses (i.e., emails exchanged), directed from the sender node to the recipient(s) node(s). The arcs are weighted by the number of emails sent from one user to the other.

The communication graph allows us to model the network and interactions by considering several well-established graph-based metrics, that represent contextual signals such as the proximity between emailers (i.e., connection strength), and larger structures within the network of individuals. One example is to measure a user's relative importance in the communication graph through her *PageRank*-score. The PageRank algorithm measures a user's (node's) relative importance through its connected arcs and their corresponding weights. In this model, each arc is a "vote of support," and thus users with a larger number of interactions receive a higher PageRank score [201].

We update the communication graph after each email that is being sent, i.e., each email either adds a new arc to the graph, or updates the weight of an existing one. We describe the utility of our communication graph for recipient prediction in Section 6.3.

6.3 Modeling

We propose a generative model that is aimed at calculating the probabilities of recipients given the sender and the content of the email. Instead of predicting a single recipient, we cast the task as a ranking problem in which we try to rank the appropriate recipients as high as possible.

More formally, let R be a candidate recipient, S the sender of an email, and E the email itself. Our final ranking will be based on the probability of observing R given S and E : $P(R | S, E)$. We use Bayes' Theorem to rewrite this probability:

$$P(R | S, E) = \frac{P(R) \cdot P(S | R) \cdot P(E | R, S)}{P(S) \cdot P(E | S)}. \quad (6.1)$$

We can explain equation 6.1 as follows: the “relevance” of a recipient is determined by (i) her prior probability (how likely is this person to receive email in general, $P(R)$ in equation 6.1), (ii) the probability of observing the sender with this particular recipient ($P(S | R)$ in equation 6.1), and (iii) the likelihood of this email to be generated from communication between the recipient and the sender ($P(E | R, S)$ in equation 6.1). To obtain the final probability, we normalize using the prior probability of the sender, and the likelihood of observing this email given its sender ($P(S) \cdot P(E | S)$ in equation 6.1).

For ranking purposes we can ignore $P(S)$ and $P(E | S)$, which will be the same for all recipients. Our final ranking function is displayed in equation 6.2.

$$P(R | S, E) \propto P(E | R, S) \cdot P(S | R) \cdot P(R). \quad (6.2)$$

In the next three sections we explain how we estimate the three components of the model: the email likelihood ($P(E | R, S)$), the sender likelihood ($P(S | R)$), and the recipient prior ($P(R)$).

6.3.1 Email Likelihood

We have several options when it comes to estimating $P(E | R, S)$. We could, for example, incorporate individual emails as latent variables. However, in this chapter we opt to directly estimate the email likelihood using the terms in the email (viz. equation 6.3).

$$P(E | R, S) = \prod_{w \in E} [\lambda P(w | R, S) + \gamma P(w | R) + \beta P(w)]. \quad (6.3)$$

In this estimation, $P(w | R, S)$ indicates the probability of observing a term w in all emails exchanged between S and R . To prevent zero probabilities, we smooth this probability with the term probability in all emails sent and received by R (i.e., $P(w | R)$ in equation 6.3) and the term probability over the whole collection (i.e., $P(w)$ in equation 6.3). In other words, we apply unigram language modeling for all the emails

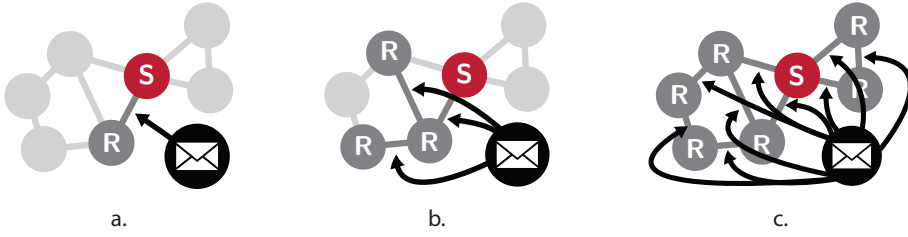


Figure 6.1: The three unigram LMs which we model from email communication. Here, the red node represents the email sender (S), the dark grey node the (candidate) recipient node(s) (R), and the light grey nodes the remaining nodes in graph G . Figure shows which emails are used to estimate $P(E | R, S)$ (a.), $P(w | R)$ (b.), and $P(w)$ (c.)

between S and R , which we smooth by the unigram LM of R (i.e., both incoming and outgoing emails), and the unigram LM of the entire background corpus (i.e., all emails in the collection up to that point in time). See also Figure 6.1.

We introduce three parameters, λ , β , and γ that represent the relative importance of each of the three components of equation 6.3, with $\lambda + \gamma + \beta = 1$, to combine the three term probabilities.

We calculate each of the three term probabilities using the maximum likelihood estimate, i.e., $P(w | \cdot) = \frac{n(w, \cdot)}{|\cdot|}$, the frequency of term w in the set of documents divided by the length of this set in number of terms.

6.3.2 Sender Likelihood

We move to the estimation of $P(S | R)$, the likelihood of observing the sender for a given recipient. Here, we use the communication graph constructed from email exchanges. The closer a recipient is in this graph and the stronger her connection to the sender S , the more likely it is that the two “belong together.” We estimate this connection strength in two ways: by considering (i) the *frequency* ($freq$), or the number of emails S sent to R at that point in time, and (ii) *co-occurrence* (co), or the number of times S and R co-occur as addressees in an email. More specifically, the frequency-based probability is defined as:

$$P_{freq}(S | R) = \frac{n(e, S \rightarrow R)}{\sum_{S' \in \mathcal{S}} n(e, S' \rightarrow R)}, \quad (6.4)$$

where $n(e, x \rightarrow R)$ indicates the number of emails sent from x to R and \mathcal{S} is the set of all senders in the graph at the current point in time. The co-occurrence-based probability is defined as:

$$P_{co}(S | R) = \frac{n(e, \rightarrow R, S)}{n(e, \rightarrow R) + n(e, \rightarrow S)}, \quad (6.5)$$

where $n(e, \rightarrow R, S)$ corresponds to the number of emails sent to *both* R and S (i.e., number of emails of which S and R are both recipients), and $n(e, \rightarrow X)$ the number of emails sent to X .

Table 6.1: Summary of Enron and Avocado email collections. We list the time span in months (Period), total number of Emails, total number of employee addresses (Addr.), the average number of emails sent ($\overline{S/p}$) and received ($\overline{R/p}$) per address.

	Period	Emails	Addr.	$\overline{S/p}$	$\overline{R/p}$
Enron	45	252,424	6,145	34	294
Avocado	58	607,011	2,068	174	321

6.3.3 Recipient Likelihood

Finally, we introduce a recipient prior, that is, the email independent probability that a recipient will be observed. This probability is unrelated to both the email at hand (E) and the sender of that email (S) and can be estimated without knowing these two variables. Again, we can choose from a variety of ways to estimate this prior probability, but we stick to two obvious choices. First, we use the *number of emails received* by R , normalized by the total number of emails sent at that point in time (*rec*). This estimation indicates how likely it is that any given email would be sent to this recipient. Second, we calculate recipient R 's *PageRank* score (*pr*) as an indication of how important R is in the communication graph.

6.4 Experimental Setup

To study the contribution of the context and content at which digital traces are created, we evaluate our recipient recommendation model using a realistic experimental setup. Our experiments aim at demonstrating the prediction effectiveness of the individual components of our model, i.e., the communication graph (CG) component and the email content component (EC), and their combination (CG+EC) as in equation 6.3. We use real enterprise email databases to evaluate our model. We optimize our models on the Enron email collection [141], and test it on the Avocado collection, which consists of email boxes of employees of an IT firm that developed products for the mobile Internet market [196]. The two collections are described in Table 6.1.

For both collections we follow the same method to select the set of users for evaluation. We first split users into three groups based on their email activity: high activity, medium activity, and low activity. This way we can study the correlation between a user's level of activity and the model's performance. As email networks typically show a long-tailed distribution [171], with a small number of users responsible for a large volume of the sent mails, and a large number of users responsible for a small volume, we define a user's activity by taking the logarithm of the number of sent emails. We prune users that have less than 100 sent mails, and compute the resulting distribution's mean (μ) and standard deviation (σ) and split the distribution into three bins: first, (i) users with *low* activity (LOW) are those that fall below $\mu - \frac{1}{2}\sigma$, (ii) *medium* active users (MED) are those between $\mu - \frac{1}{2}\sigma$ and $\mu + \frac{1}{2}\sigma$, and finally, (iii) *highly* active users (HIGH) are the ones over $\mu + \frac{1}{2}\sigma$. From each bin we randomly sample 50 users, which results in our final evaluation set of 150 users.

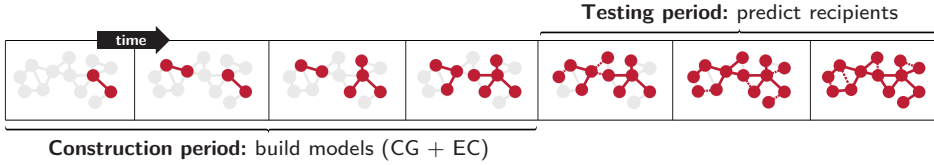


Figure 6.2: Our evaluation methodology, where we have an initial *construction period* to generate language models and build the communication graph. During the subsequent *testing period* we select emails for which predict recipients (represented by the dotted lines). We continue building the communication graph and updating the language models in the testing period.

6.4.1 Evaluation

Before we start predicting email recipients we allow our model to gather evidence from all email communication up to that point. More specifically, we use an initial *construction period* to generate the users' language models and the communication graph. We start to predict recipients in the subsequent *testing period*. During both periods our model is updated for each sent email. We split each user's period of activity (starting from the user's first sent email, up to the last sent mail) into the *construction period*, covering $\frac{2}{3}$ of the emails, and the *testing period*, which is $\frac{1}{3}$ of the emails. See also Figure 6.2).

For each sender in our user evaluation set, we select 10 emails, evenly distributed over the testing period as evaluation points. For each of these emails we rank the top 10 recipients and compare to the actual recipients of the email. We report on mean average precision (MAP), as it allows us to identify improvements in the ranking of recipients. We indicate the best performing model in boldface and test statistical significance using a two-tailed paired t -test. Significant differences are marked $\blacktriangle/\blacktriangledown$ for $\alpha = 0.01$ and \triangle/\triangledown for $\alpha = 0.05$.

6.4.2 Parameter Tuning

We use the Enron collection to tune the parameters λ , γ , and β in equation 6.3, and to determine which methods for estimating the sender ($P(S | R)$) and recipient ($P(R)$) likelihood work best. The results of the parameter tuning are displayed in Tables 6.2 and 6.3. The final settings we use for testing our model on the Avocado collection are the following: $\lambda = 0.6$, $\gamma = 0.2$, $\beta = 0.2$, we estimate $P(S | R)$ using the number of emails S sent to R , $P_{freq}(S | R)$, and we estimate $P(R)$ using the number of emails received by R , $P_{rec}(R)$.

6.5 Results and Analysis

We compare the found optimal settings for the CG and EC components to their combination using our training collection (Enron). Table 6.4 shows that our hybrid model significantly outperforms either of the single models across all groups of users in the Enron collection, even if the performance increase is modest. The highest performance is

Table 6.2: System performance (MAP) on the Enron dataset over different methods for estimating $P(R)$ and $P(S | R)$ (Section 6.3).

	$P_{pr}(R),$ $P_{co}(S R)$	$P_{pr}(R),$ $P_{freq}(S R)$	$P_{rec}(R),$ $P_{co}(S R)$	$P_{rec}(R),$ $P_{freq}(S R)$
LOW	0.2207	0.1488	0.2317	0.4365
MED	0.1961	0.2334	0.2116	0.3857
HIGH	0.1016	0.1213	0.1169	0.2060
ALL	0.1755	0.1676	0.1893	0.3480

Table 6.3: System performance (MAP) on the Enron dataset over a parameter sweep for the parameters λ , γ , and $\beta = 1 - (\lambda + \gamma)$ in equation 6.3 with a step size of 0.2.

$\lambda \downarrow / \gamma \rightarrow$	0.2	0.4	0.6
0.2	0.4670	0.4699	0.4752
0.4	0.5070	0.5095	
0.6	0.5258		

achieved in the LOW and MED user groups: lower user activity correlates positively with performance.

We present the results of our final experiments on the Avocado set in Table 6.4. Compared to the results on the Enron set, which we used for tuning our parameters, our model’s performance is higher throughout on the Avocado set, both across the different models, and within each subgroup of users. An indication for this difference in absolute performance scores comes from the collection statistics in Table 6.1. Here we see that the Avocado set contains fewer unique addressees, spans a longer period of time, and contains a larger number of emails per person. As a possible factor contributing to our models’ higher performance on the Avocado collection, we point to the fact that our model has more data available to leverage for ranking a smaller number of candidates. While different datasets may need different models, the consistently high scores show that the components work in isolation and in combination over different datasets.

6.5.1 Between Groups Comparison

Similar to what we saw in the experiment on the Enron collection, higher user activity generally seems to result in lower performance on the Avocado dataset (Table 6.4). The CG model is an exception and outperforms our combined model for highly active users. The combined model achieves significant performance improvements over the content model in each subset of users.

To better understand these patterns, we turn to the characteristics of the different user subgroups. We plot the users’ numbers of emails (both sent and received), indicating their activity, and juxtapose it to the size of their *egonet*, which corresponds to the set of directly connected neighbors in the communication graph [5]. This *egonet* represents the

Table 6.4: System performance (MAP) on Enron and Avocado. We test for statistically significant differences of the individual models (CG and EC) against the combined model (CG+EC).

	Enron			Avocado		
	CG	EC	CG+EC	CG	EC	CG+EC
LOW	0.4365 [▼]	0.5757 [▼]	0.5833	0.6502 [▼]	0.6946 [▼]	0.7077
MED	0.3857 [▼]	0.5161 [▼]	0.5325	0.6052 [▽]	0.6328 [▼]	0.6542
HIGH	0.2060 [▼]	0.4779 [▽]	0.4853	0.6652[△]	0.5739 [▼]	0.6136
ALL	0.3480 [▼]	0.5258 [▼]	0.5362	0.6402	0.6352 [▼]	0.6597

Table 6.5: Trend line slopes after applying linear regression to the individual and combined components data.

Model	Slope
Email Content (EC)	$1.36 \cdot 10^{-4}$
Communication Graph (CG)	$-1.38 \cdot 10^{-4}$
Combined (EC+CG)	$1.27 \cdot 10^{-4}$

users they interact(ed) with, and is indicative of a user’s reach or embedding inside the communication graph.

The resulting plot is shown in Figure 6.3. There is a clear clustering: highly active users (orange squares) who send and receive a large number of emails, also have a larger number of people they interact with. While more textual content allows the generative model to create richer recipient profiles, in turn enabling more informed recipient ranking, there is a catch to a larger egonet too. The sender-recipient communication smoothing in our generative model results in a larger number of high-scoring candidates for highly active users. This makes it more difficult for the ranker to discern the true recipient(s) from the larger pool.

6.5.2 Performance over Time

Motivated by the fact that our model is updated at each sent email, we study its performance over time. To get an indication of whether our model’s performance improves or deteriorates over time, we apply linear regression on the data points of each model, and fit a trend line. See Table 6.5 for the trend line slopes for each individual component, and the combined model.

Both the EC and combined model’s trend lines have positive slopes, whilst the CG model has a negative slope. This indicates that our language modeling approach benefits from the larger amount of textual content it receives for each recipient over time, allowing the generation of richer recipient profiles for better email likelihood estimations. The CG approach on the other hand, deteriorates over time, suffering from the increased size and complexity of the communication graph. We note that, as time progresses, the

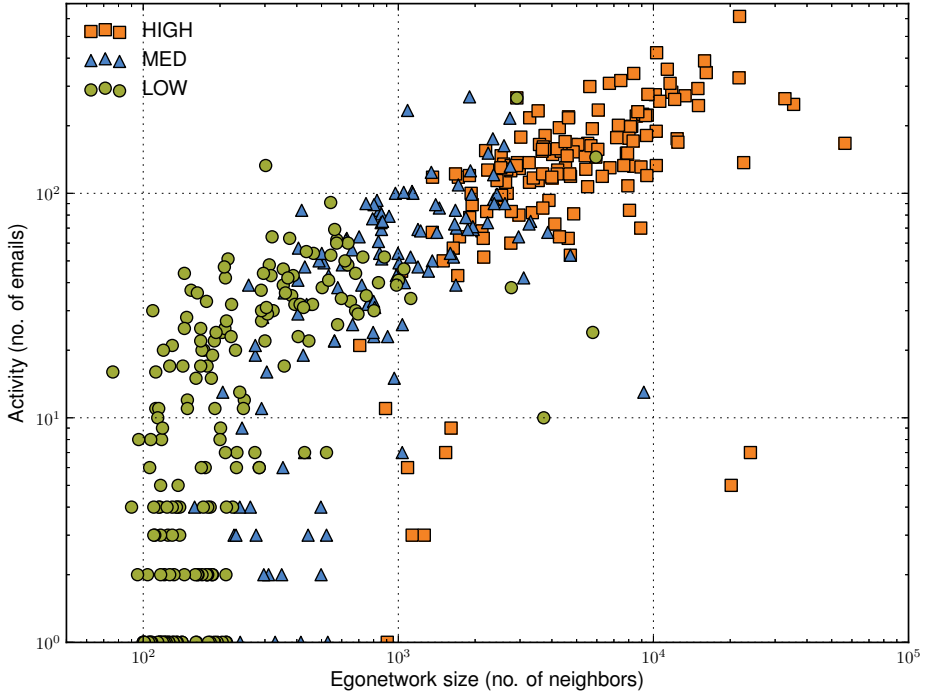


Figure 6.3: The three user groups in Avocado, showing each user’s activity (y-axis) against the size of its egonetwork (x-axis).

communication graph model “settles in,” and becomes less likely to pick up on changing balances or shifting communication patterns in the communication graph. For future work we argue for a time-aware model, that is able to adapt to shifts in the communication graph over time, by taking recency into account.

6.5.3 Rankers’ Correlation

To analyze the performance of our combined model in comparison to the isolated ones, we look at their rankings and compute the Kendall tau rank correlation coefficient (τ) between pairs of models. The top plot in Figure 6.4 shows how agreement between CG and EC is relatively low, centering around the 0 mark with an average of 0.0471. This pattern largely coincides with that of the second plot, which depicts agreement between CG and our combined model. The average correlation coefficient is only slightly higher at 0.0562. Finally, the agreement is highest between the EC model and our combined model, at on average 0.7425. The high agreement offers an explanation for the comparatively low performance of our combined model in the HIGH subset of the Avocado set. The EC model’s comparatively low performance (as seen in Table 6.4), indicates that the combined model is negatively affected by following EC’s incorrect rankings.

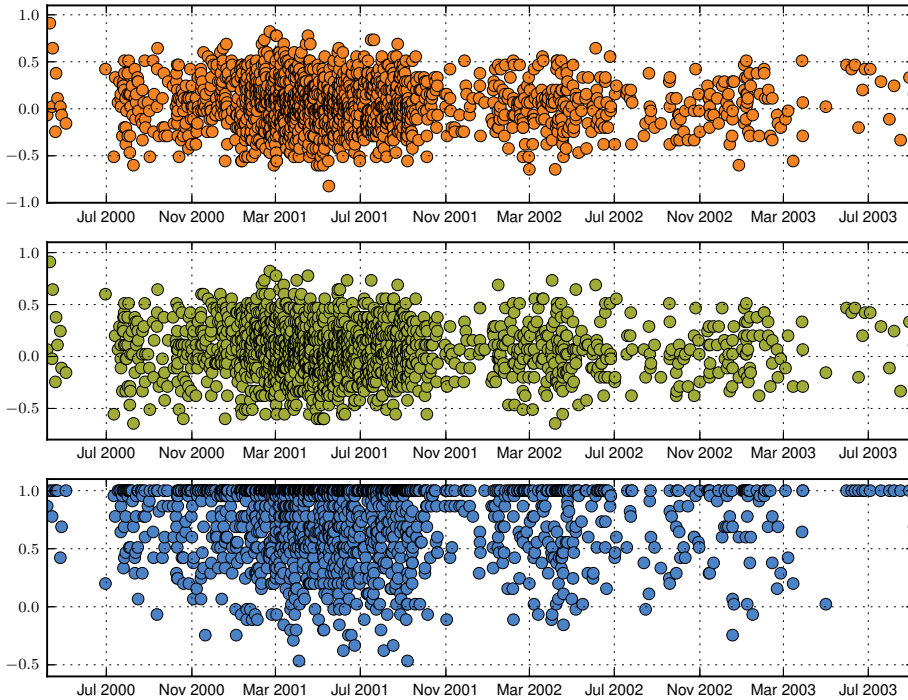


Figure 6.4: Kendall's τ over time, between CG and EC (orange), CG and CG+EC (green), and EC and CG+EC (blue).

6.6 Conclusion

In this chapter, we presented a novel hybrid model for email recipient prediction that leverages both the information from an email network's communication graph and the textual content of the messages. Our model starts from scratch, in that it does not assume or need seed recipients, and it is updated for each email sent. The proposed model achieves high performance on two email collections. We have answered RQ4:

RQ4 Can we predict email communication through modeling email content and communication graph properties?

We show that the communication graph properties and the email content signals both provide a strong baseline for predicting recipients, but are complementary, i.e., combining both signals achieves the highest accuracy for predicting the recipients of email.

We have shown that the number of received emails is an effective method for estimating the prior probability of observing a recipient, and the number of emails sent between two users is an effective way of estimating the “connectedness” between these two users, and proves to be a helpful signal in ranking recipients.

An implication of the findings in this chapter is that both the *context* in which digital traces are created, and the *content* of the digital traces are valuable signals in predicting

real-world behavior, in the case of email communication. This has implications for both designing systems to predict recipients, as it does for analyzing and understanding communication flows in an enterprise, e.g., in a discovery scenario where one may want to discover “unexpected” communication.

The work presented in this chapter has several limitations, first, from a modeling perspective; We identified characteristic weaknesses of our individual models’ robustness to specific circumstances. As witnessed by the decrease in performance for highly active users, our email content model seems unfit to deal with users that exchange a large number of emails with a large number of people. The deteriorating performance over time shows that our communication graph models’ performance suffers with expanding graphs as they develop.

To address these issues, one possible direction for future work is to incorporate time into our model. We can, for example, use decay functions to weigh the edges between users and promote more recent communication. Similarly, we can use time-dependent language models that favor recent documents to incorporate time. Whereas in this work we choose to keep a simple, and hence interpretable model, a machine learned method may improve prediction accuracy, e.g., by using the signals we employed as features. Finally, for future work, the presented model can be applied to novel tasks, e.g., role or anomaly detection in email corpora, or leak prevention.

In the next chapter, we take a similar approach to the work presented in this chapter. More specifically, we analyze a novel dataset of interaction logs, and address a prediction task. Much like the recipient recommendation task in this chapter, we study the contribution of the context and the content of the digital traces, with the goal of predicting real-world activity of our entities of interest.

Analyzing and Predicting Task Reminders

“The advantage of a bad memory is that one enjoys
several times the same good things for the first time.”

—*Friedrich Nietzsche*

Our second case-study into the link between digital traces and real-world behavior or activities of entities of interest, revolves around mobile devices. Mobile devices are an especially interesting source of digital traces, as the closeness, portability, and popularity of our smartphones means they are part of most of the aspects of our daily lives. Mobile device hence take a central role as potential “containers for evidence” in the E-Discovery scenario [40]. Smartphones may carry signals of our whereabouts, context, and activities in many different forms, e.g., from emails, (voice) messages, call and SMS logs, to browser data, and other data logs [16].

A relatively new source of digital traces that can provide rich insights into users’ activities and behavior are automated personal assistants. These intelligent assistants have been introduced in major online service offerings and mobile devices over the last several years. Systems such as Siri, Google Now, Echo, M, and Cortana support a range of reactive and proactive scenarios, ranging from question answering to alerting about plane flights and traffic. The embedding of these personal assistants in our day-to-day life makes them a rich potential resource of “digital evidence,” i.e., digital traces that can be employed for finding evidence of activity in the real world [197].

Several of these personal assistants provide reminder services aimed at helping people to remember plans for future tasks they may otherwise forget. Potentially representing the user’s (future) location, activity, or planned tasks, interaction logs with these types of reminder services could serve as a valuable signal in an E-Discovery scenario. In this chapter, we study a large-scale log of user-created reminders from Microsoft Cortana, to better understand the context under which digital traces are produced, and to study whether we can predict the (planned) behavior of the *entities of interest* of Part II: the producers of digital traces.

Motivated by the findings in the previous chapter, in this chapter too we study the impact of the *contexts* in which the traces are created. Here, we consider the creation time of a reminder as the context. In addition, we study the predictive power of the textual *content* of the digital traces, by incorporating the reminder text for predicting the reminder task’s execution time.

Table 7.1: Example interaction sequence for setting a reminder.

Turn	Who	Text
1	User	Remind me to do the laundry.
2	System	When would you like to be reminded?
3	User	Sunday at noon.
4	System	Alright, remind you to do the laundry at 12:00PM on Sunday, is that right?
5	User	Yes.
6	System	Great, I'll remind you! success chime

Table 7.1 presents an example of the types of reminder dialogs recorded in the dataset. These logs offer insights into the reminder generation process, including the types of tasks for which people formulate reminders, the task descriptions, the times that reminders of different types are created, and the periods of time between the creation of reminders and notifications. Beyond analysis of the nature and timing of reminders, we demonstrate how information about patterns of reminder usage and general trends seen across users can be harnessed to predict when users are likely to execute certain tasks, which can be of interest in an E-Discovery scenario. We focus primarily on time-based reminders, i.e., reminders for tasks planned for a future time. In this chapter, we aim to answer RQ5: “*Can we identify patterns in the times at which people create reminders, and, via notification times, when the associated tasks are to be executed?*”

7.1 Reminder Types

We first investigate user behavior around reminder creation by studying the common tasks linked to setting reminders. In this section, we focus on the question: “*Is there a body of common tasks that underlie the reminder creation process?*” Understanding the common tasks allows us to better see common usage patterns, which will help in predictive models and in finding unexpected behavior, e.g., through outlier detection. To answer this question, we perform a data-driven qualitative analysis of Cortana reminder logs. Specifically, we extract reminders that are observed frequently and across multiple users. Then, we employ a manual labeling strategy, and categorize the reminders in a task taxonomy to better understand the task types that drive users to create reminders.

7.1.1 Reminder Composition

Before we analyze the different tasks at the root at reminder setting, we describe the composition of a typical reminder. In the left column of Table 7.2, we present three examples of common reminders. The examples show a structure that is frequently observed in the logged reminders. Reminders are typically composed as predicate sentences. They contain a phrase related to an action that the user would like to perform (typically a verb phrase) and a referenced object that is the target of the action to be performed.

Table 7.2: Example reminders as predicates.

Reminder	Predicate
“Remind me to take out the trash”	Take out (me, the trash)
“Remind me to put my clothes in dryer”	Put (me, clothes in dryer)
“Remind me to get cash from the bank”	Get (me, cash from the bank)

7.1.2 Data

A session for setting a reminder consists of a dialog, where the user and the intelligent assistant interact in multiple turns. Typically, the user starts by issuing the command for setting a reminder, and dictates the reminder. Optionally, the user specifies the reminder’s notification time. Next, the assistant requests to specify the time (if the user has not yet specified it), or provides a summary of the reminder, i.e., the task description and notification time, asking the user to confirm or change the proposed reminder (see Table 7.1).

We analyze a sample of two months of Cortana reminder logs, spanning all of January and February 2015. We ensure the data used in this chapter preserves user privacy, as we filter the data to patterns commonly observed across multiple users, and study behavior in aggregate, refraining from studying individual patterns. In summary, we are not looking at any user’s behavior at the individual level, but across a large population, to uncover broad and more general patterns.

We pre-process this set of reminders by including only reminders from the US market (the only market which had Cortana enabled on mobile devices at that time). To narrow the scope of our analysis, we focus on time-based reminders and remove location (e.g., “*remind me to do X when I am at Y*”) and person-based reminders (e.g., “*remind me to give X when I see Y*”), which are less common and more challenging to study across users due to their personal nature. Finally, we retain only reminders that are confirmed by the user (turn 6 in Table 7.1). The resulting sample contains 576,080 reminders from 92,264 users. For each reminder, we extract the reminder task description and notification time from Cortana’s summary (turn 4 in Table 7.1). We also extract the creation time based on the local time of the user’s device. Each reminder is represented by:

r_{task} The textual task description of the reminder, i.e., the phrase which encodes the future task or action to be taken, as dictated by the user. We extract the text from Cortana’s final summary response (“do the laundry” from turn 4 in Table 7.1).

r_{CT} The creation time of the reminder. This represents the time at which the user encodes the reminder. We represent r_{CT} as a discretized time-value; Section 7.3.1 defines the discretization process. We extract this timestamp from the client’s device.

r_{NT} The notification time set for the reminder to fire an alert. This data represents the time at which the user wishes to be reminded about a future task or action. We represent r_{NT} in the same discretized manner as r_{CT} . We extract the notification time from Cortana’s summary response (turn 4 in Table 7.1).

$r_{\Delta T}$ Subtracting the creation time from the notification time yields the time-delta, the delay between the reminder’s creation and notification time. Intuitively, reminders with smaller time-deltas represent short-term or immediate tasks (“*remind me to take the pizza out of the oven*”), whereas reminders with larger time-deltas represent tasks planned further ahead in time (“*remind me to make a doctor’s appointment*”).

7.1.3 Identifying Common Tasks

To understand the common needs that underlie the reminder creation process, and hence common user behavior, we first identify common reminders, i.e., reminders that are frequently observed across multiple users. Studying common reminders can aid analysts in understanding common usage patterns, and hence can help in identifying patterns or reminders that diverge. In addition, common usage patterns can aid system designers in creating predictive models. We employ a mixed methods approach, comprising data-driven and qualitative methodologies, to extract and identify common task types.

Frequent Task Description Extraction

First, we extract common task descriptions, by leveraging the predicate (verb+object) structure described at the start of this section. To ensure the underlying task descriptions represent broad tasks, we filter to retain only descriptions that start with a verb (or a multi-word phrasal verb) that occurs at least 500 times, across at least ten users, with at least five objects. This yields a set of 52 frequent verbs, which covers 60.9% of the reminders in our sample. The relatively small number of verbs that cover the majority of reminders in our log indicates there likely exists a large ‘head’ of common tasks that give rise to reminder creation. To analyze the underlying tasks, we include the most common objects, by pruning objects observed less than five times with a verb. This yields a set of 2,484 unique task descriptions (i.e., verb+object pairs), covering 17.9% of our sample log.

Manual Labeling

Next, we aim to identify the tasks which underlie the frequent task descriptions, and categorize them into a task type taxonomy. Specifically, by manual inspection, we identified several key dimensions that separate tasks. In particular, we find that dimensions that commonly separate tasks are (i) the cognitive load the task incurs (i.e., how much does the task represent an interruption of the user’s activity), (ii) the context in which the task is to be executed (i.e., at home, at work), and (iii) the (expected) duration of the task. This enabled us to categorize the extracted frequent task descriptions into one of six broader task types, with several subclasses, which we present in the following section with example reminders.

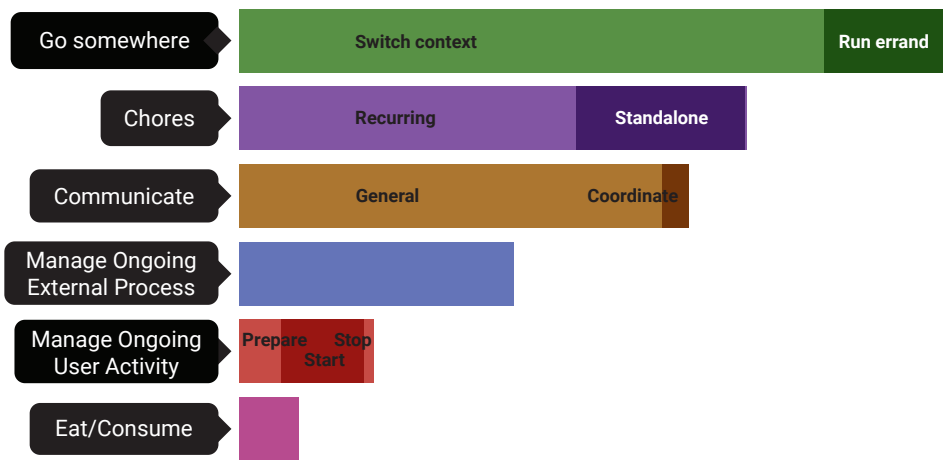


Figure 7.1: Bar plot showing the proposed reminder task-type taxonomy.

7.2 Task Type Taxonomy

In this section, we describe each of the six task types in turn, and provide examples of the associated verb+object patterns. First, for an overview of the task types and their relative occurrence in our data, see Figure 7.1. The example objects are shown in decreasing order of frequency, starting with the most common. Note that verbs are not uniquely associated with a single task type, but the verb+object-pair may determine the task type (compare, e.g., “set alarm” to “set doctor appointment”).

1. Go Somewhere (33.0%)

One third of the frequent tasks refer to the user moving from one place to another. See Table 7.3. We distinguish between two subtypes: the first subtype is running an errand (83.2%), where the reminder refers to executing a task at some location (e.g., “pick up milk”). Running an errand represents an interruption of the user’s activity, but a task of a relatively small scale, i.e., it represents a task that briefly takes up the user’s availability. The second subtype is more comprehensive, and represents tasks that are characterized by a switch of context (16.8%), e.g., moving from one context or activity to another (“go to work”, “leave for office”), which has a larger impact on the user’s availability.

Table 7.3: Go Somewhere.

Run errand	
grab [something]	laundry, lunch, headphones
get [something]	batteries
pick up [something/someone]	laundry, person, pizza
buy [something]	milk, flowers, coffee, pizza
bring [something]	laptop, lunch, phone charger
drop off [something]	car, dry cleaning, prescription
return [something]	library books
Switch context	
leave (for) [some place]	house, work, airport
come [somewhere]	home, back to work, in
be [somewhere]	be at work, at home
go (to) [somewhere]	gym, work, home, appointment,
stop by [some place]	the bank, at Walmart
have (to) [something]	work, appointment

2. Chores (23.8%)

The second most common type of reminders represent daily chores. See Table 7.4. We distinguish two subtypes: recurring (66.5%) and standalone chores (33.5%). Both types represent smaller-scale tasks that briefly interrupt the user’s activity.

Table 7.4: Chores.

Recurring	
take out [something]	trash, bins
feed [something]	dogs, meter, cats, baby
clean [something]	room, house, bathroom
wash [something]	clothes, hair, dishes, car
charge [something]	phone, fitbit, batteries
do [something]	laundry, homework, taxes, yoga
pay [something]	pay rent, bills, phone bill
set [something]	alarm, reminder
Standalone	
write [something]	a check, letter, thank you note
change [something]	laundry, oil, air filter
cancel [something]	amazon prime, netflix
order [something]	pizza, flowers
renew [something]	books, driver’s license, passport
book [something]	hotel, flight
mail [something]	letter, package, check
submit [something]	timesheet, timecard, expenses
fill out [something]	application, timesheet, form
print [something]	tickets, paper, boarding pass
pack [something]	lunch, gym clothes, clothes

3. Communicate (21.1%)

Next, a common task is to remind to contact (“*call*,” “*phone*,” “*text*”) another individual, either a person (e.g., “*mom*,” “*jack*,” “*jane*”), organization/company (“*AT&T*”), or other (“*hair dresser*,” “*doctor’s office*”). See Table 7.5. We identify two subtypes: the majority reflects general, unspecified communication (94.7%) (e.g., “*call mom*”), and a smaller part (5.3%) represents coordination or planning tasks (e.g., “*make doctor’s appointment*”). Both subtypes represent tasks that briefly interrupt the user’s activity.

Table 7.5: Communicate	
General	
send [something]	email, text, report
email [someone]	dad, mom
text [someone]	mom, dad
call [someone]	mom, dad
tell [someone] [something]	my wife I love her, happy birthday mom
Coordinate	
set [an appointment]	doctors appointment
make [an appointment]	doctors appointment, reservation
schedule [an appointment]	haircut, doctors appointment

4. Manage Ongoing External Process (12.9%)

These reminders represent manipulation of an ongoing, external process, i.e., tasks where the user monitors or interacts with something, e.g., the laundry or oven. See Table 7.6. These tasks briefly interrupt a user’s activity.

Table 7.6: Manage Ongoing External Process	
turn [on/off] [something]	water, oven, stove, heater
check [something]	email, oven, laundry, food
start [something]	dishwasher, laundry
put [something] in [something]	pizza in oven, clothes in dryer
take [something] out	pizza, chicken, laundry

5. Manage Ongoing User Activity (6.3%)

This class of reminders is similar to the previous class, however, as opposed to the user interacting with an external process, these reflect a change in the user's own activity. They incur a higher cost on the user's availability and cognitive load. See Table 7.7. We distinguish three subtypes: preparing (31.4%), starting (61.4%), and stopping an activity (7.2%).

Table 7.7: Manage Ongoing User Activity

Activity/Prepare	
get ready [to/for]	work, home
Activity/Start	
start [some activity]	dinner, cooking, studying
make [something]	food, breakfast, grocery list
take [something]	a shower, a break
play [something]	game, xbox, basketball
watch [something]	tv, the walking dead, seahawks game
Activity/Stop	
stop [some activity]	reading, playing
finish [something]	homework, laundry, taxes

6. Eat/Consume (2.8%)

Another frequent reminder type refers to consuming something, most typically food (“*have lunch*”) or medicine (“*take medicine*”). These are also small and range from brief interruptions (“*take pills*”) to longer interruptions (“*have dinner*”). See Table 7.8.

Table 7.8: Eat/Consume

take [something]	Medicine
eat [something]	lunch, dinner, breakfast, pizza
have [something]	lunch, a snack, breakfast

7.3 Reminder Patterns

Next, we study the temporal patterns of reminders. We seek to understand when people create reminders, when reminders are set to notify users, and the average delay between creation and notification time for different reminders. Such knowledge could prove useful in providing likelihoods about when certain tasks tend to happen or predicting (follow-up) tasks. In this section, we focus on the research question: *“Can we identify patterns in the times at which people create reminders, and, via notification times, when the associated tasks are to be executed?”*

We study reminders on several levels of granularity. In Section 7.3.2 we look at global patterns and trends across all reminders. Next, we study temporal patterns per task type in Section 7.3.3. In Section 7.3.4, we perform a temporal analysis of task description terms. Finally, we study the relation between reminder creation and notification times in Section 7.3.5. First, we explain how we represent the reminder’s creation and notification time to enable our analyses.

7.3.1 Method

To study common temporal patterns of reminders, we discretize time by dividing each day into the following six four-hour buckets: (i) late night [00:00-04:00), (ii) early morning [04:00-08:00), (iii) morning [08:00-12:00), (iv) afternoon [12:00-16:00), (v) evening [16:00-20:00), and (vi) night [20:00-00:00). By combining this time-of-day division with the days of week we yield a 7 by 6 matrix M , whose columns represent days, and rows times. Each r_{CT} and r_{NT} can be represented as a cell in matrix M , i.e., $M_{i,j}$ where i corresponds to the day of week, and j to the time of day. Furthermore, we distinguish between M^{CT} and M^{NT} , the matrices whose cells contain reminders that are created, or respectively set to notify, at a particular day and time. We represent each reminder as an object, with the attributes described in Section 7.1.2: the reminder’s task description (r_{task}), creation time (r_{CT}), notification time (r_{NT}), and time-delta ($r_{\Delta T}$). To study the temporal patterns, we look at the number of reminders that are created, or whose notifications are set, per cell. We compute conditional probabilities over the cells in M^{CT} and M^{NT} , where the reminders creation or notification time is conditioned on the task type, time, or the terms in the task description.

$$P(r_{CT} \mid w) = \frac{|\{r \in R : w \in r_{task} \wedge r_{CT} = X\}|}{|\{w \in r_{task}, r \in M^{CT}\}|} \quad (7.1)$$

$$P(r_{NT} \mid w) = \frac{|\{r \in R : w \in r_{task} \wedge r_{NT} = X\}|}{|\{w \in r_{task}, r \in M^{NT}\}|} \quad (7.2)$$

To estimate the conditional probability of a notification or creation time, given a term from the task description, we take the set of reminders containing term w that are created or whose notification is set at time X , over the total number of reminders that contain the word (see equation 7.1 and equation 7.2). By computing this probability for each cell in M^{NT} or M^{CT} , (i.e., $\sum_{i,j \in M} P(r_{NT} = i, j \mid w)$) we generate a probability distribution over matrix M .

$$P(r_{NT} = X \mid r_{CT} = i, j) = \sum_{i, j \in M^{CT}} \frac{|\{r \in M_{i,j}^{CT} : r_{NT} = X\}|}{|M_{i,j}^{CT}|} \quad (7.3)$$

To study the common patterns of the periods of time between the creation of reminders and notifications, we estimate a probability distribution for a reminder's notification time given a creation time (see equation 7.3). We compute this probability by taking the reminders in each cell of M^{CT} that have their notifications set to fire at time X , over all the reminders in that cell.

Finally, we study the delays between setting and executing reminders, by collecting counts and plotting histograms of $r_{\Delta T}$ of reminders for a given subset, e.g., $\{r_{CT} \in R : w \in r\}$ or $\{r_{CT} \in R : r_{CT} = X\}$.

7.3.2 Global Patterns

We now describe broad patterns of usage, and answer the following questions: “*At which times during the day do people plan (i.e., create reminders), and at which times do they execute tasks (i.e., reminder notification trigger)?*” and “*How far in advance do people plan tasks?*” To answer these questions, we examine the temporal patterns in our log data over the aggregate of all reminders in the two-month sample (576,080 reminders).

Figure 7.2 shows the prior probability of a reminder's creation time, $P(r_{CT})$, and notification time, $P(r_{NT})$, in each cell in M^{CT} and M^{NT} . Looking at Figure 7.2, we see that in our sample, planning (reminder creation) most frequently happens later in the day, more so than during office hours (morning and midday). This observation could be explained by the user's availability; users may have more time to interact with their mobile devices in the evenings. Additionally, the end of the day is a natural moment for “wrapping up the day,” i.e., looking backward and forward to completed and future tasks.

Turning our attention to notification times, the right plot of Figure 7.2 shows a slightly different pattern: people execute tasks (i.e. notifications trigger) throughout the day, from morning to evening. This shows that users want to be reminded of tasks throughout the day, in different contexts (e.g., both at home and at work). This is reflected in our task-type taxonomy, where tasks are related to both contexts. We also note how slightly more notifications trigger on weekdays than in weekends, and more notifications trigger at the start and end of the workweek than midweek and in weekends. This observation may be attributed to the same phenomenon for reminder creation; users may tend to employ reminders for activities that switch between week and weekend contexts. Finally, comparing the two plots shows the notification times are slightly less uniformly distributed than creation times, e.g., users create reminders late at night, when it is relatively unlikely for notifications to fire.

Next, to determine how far in advance users typically plan, we look at the delays between reminder creation and notification in Figure 7.3. The top plot shows distinct spikes around five-minute intervals, which are due to reminders with a relative time indication (e.g., “*remind me to take out the pizza in 5 minutes*”). These intervals are more likely to come to mind than more obscure time horizons (e.g., “*remind me to take out the pizza at 6.34pm*”). The second and third plots clearly illustrate that the majority of reminders have a short delay: around 25% of the reminders are set to notify within the

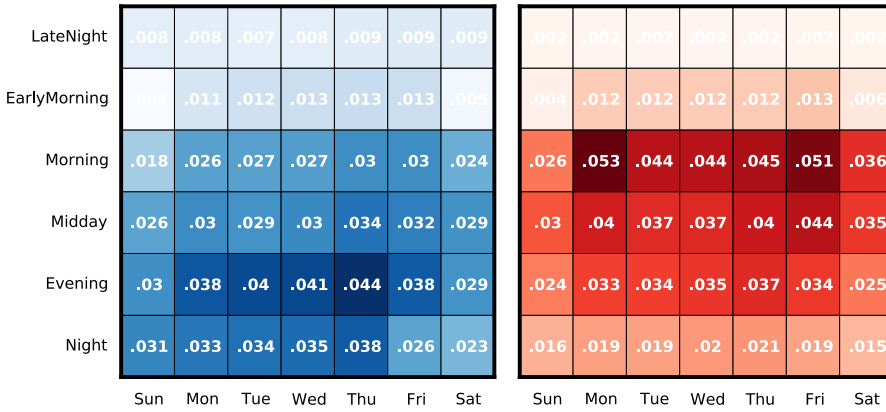


Figure 7.2: Distribution of reminder creation times (left plot) and reminder notification times (right plot) for all reminders in two-month sample ($n = 576,080$)

same hour (second plot), and around 80% of the reminders are set to notify within 24 hours (third plot). Interestingly, there is a small hump around 8–9 hours in the second plot, which may be explained by reminders that span a night, e.g., created at the end of the day, to notify early the next day, or a ‘working day’ (reminder creation in the morning, notification at the end of the day).

In summary, we have shown that on average people tend to set plans in the evening, and execute them throughout the day. Furthermore, most tasks that drive reminder setting are for short-term tasks to be executed in the next 24 hours.

7.3.3 Temporal Patterns of Different Task Types

In this section, we explore whether different task types are characterized by distinct temporal patterns that differ from the global patterns seen in the previous section. To do so, we take the subset of the 2,484 unique frequent reminders in our two-month sample of reminders. We extract all reminders that match the exact reminder task descriptions from our full dataset, and yield a subset of 125,376 reminders with task type-labels, that we use for analysis. We aim to answer the same questions raised in the previous section, but at the level of task type, as opposed to a characterization of the global aggregate.

Creation and notification times. First, we look at the probability distribution of reminder creation times per task type, i.e., $P(r_{CT} \mid tasktype)$. Looking at the distributions for each task type, we discover two broader groups: per task type, reminders are either created mostly in morning and midday blocks (roughly corresponding to office hours), or outside these blocks. Figure 7.4 shows examples of both types: “Activity” and “Going somewhere” reminders are mostly created during office hours, while e.g., “Communicate” and “Chore” reminders are more prone to be created in evenings. Another interesting observation is that activity-related reminders are comparatively frequent on weekends.

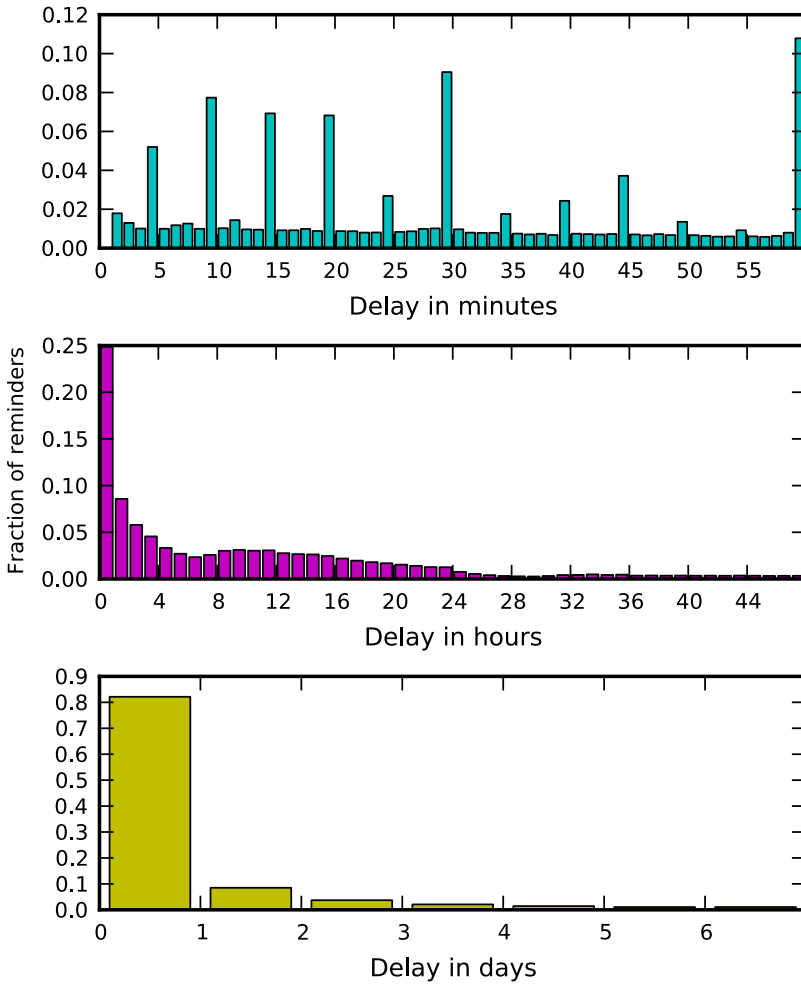


Figure 7.3: Histograms of delays (in minutes, hours, and days from top to bottom) between reminder creation and notification times.

Next, we study reminder notification times per task type, i.e., $P(r_{NT} \mid \text{tasktype})$. Here, a similar pattern emerges. Broadly speaking, there are two types of tasks: those set to notify during office hours, and those that trigger outside these hours. See Figure 7.5 for examples. “Communicate” and “Go” fall under the former type, whereas “Chore” and “Manage ongoing process” fall under the latter. The nature of the tasks explains this distinction: the former relate to work-related tasks (communication, work-related errands), whilst the majority of the latter represent activities that are more common in a home setting (cooking, cleaning).

Taking a closer look at the Communicate task subclasses in Figure 7.6, we show

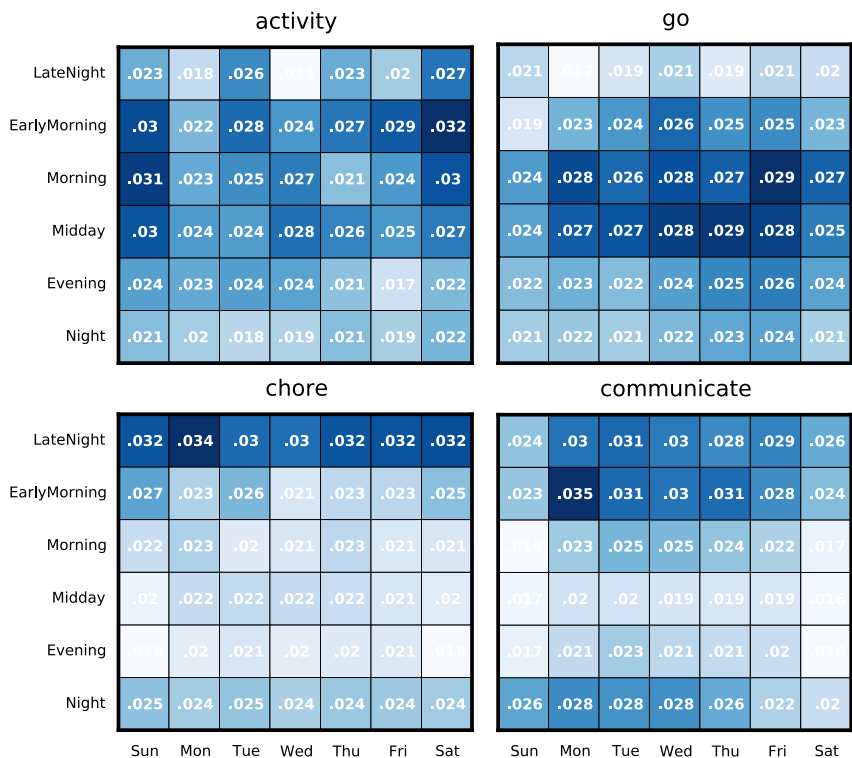


Figure 7.4: Reminder creation time probability distributions over time, for different task types.

how “Communicate/General” and “Communicate/Coordinate” differ: the former is more uniformly distributed, whilst the latter is denser around office hours. The general subclass too has comparatively more reminders that trigger in weekends, whereas coordinate is more strongly centered on weekdays. The distinct patterns suggest these subclasses indeed represent different types of tasks.

Reminder creation and notification delay. To better understand differences in the lead times between reminder creation and notification, we present an overview of the distribution of reminder delays per task type in Figure 7.7. In general, the lower the boxplot lies on the y -axis, the lower the lead time, i.e., the shorter the delay between creating the reminder and executing the task. It is worth comparing, e.g., the plot of “Manage ongoing process,” to both “Go” or “Communicate” task types: execution of managing ongoing processes tasks seems to be planned with a much shorter lead time than the other types of task. Considering the nature of the tasks, where ongoing processes often represent the close monitoring or checking of a process (e.g., cooking or cleaning tasks), it is understandable that the delays are on the order of a few minutes, rather than

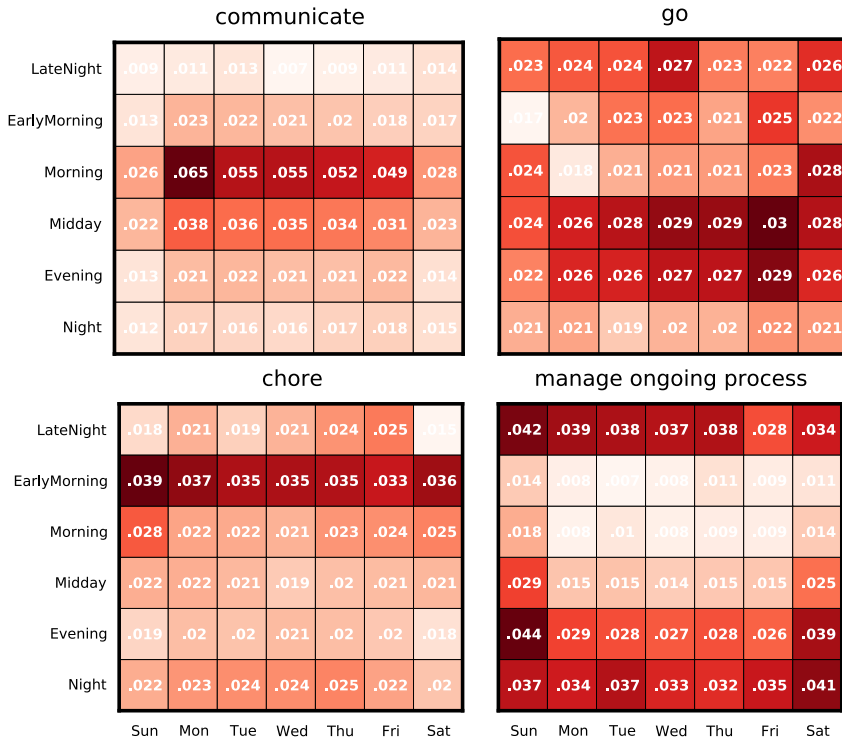


Figure 7.5: Reminder notification time probability distributions over time, per task type.

hours. “Communicate/Coordinate” has the largest delay on average, i.e., it is the task type people plan furthest in advance.

A more detailed examination of the differences between the “Communicate” subtasks, illustrated in Figure 7.8, reveals that “Communicate/General” subtasks are more likely to be executed with lower lead time, as noted by the peak at hour 0 in the top plot. The “Communicate/Coordinate” subtask is about as likely to be executed the next day, as seen by the high peak around the 12 hour mark in the bottom plot. Much like the observations made in the previous section, the difference in the patterns between both “Communicate” subtasks suggests that the distinction between the subtypes is meaningful. Differences are not only found on a semantic level through our qualitative analysis, but also in temporal patterns.

In summary, we have shown how task type-specific temporal patterns differ from the aggregate patterns in Section 7.3.2.

7.3.4 Temporal Patterns of Terms in Reminder Task Descriptions

One can hypothesize that the terms in task descriptions show distinct temporal patterns, i.e., reminders that contain the term “pizza” are likely to trigger around dinner time.

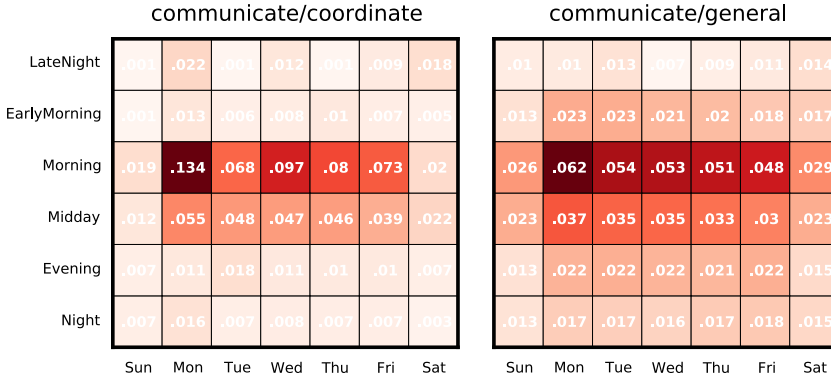


Figure 7.6: “Communicate” subclass notification time probability distributions over time.

Presence of these temporal patterns may be leveraged for reminder creation or notification time prediction. To study this, we manually inspected the temporal distribution of task descriptions’ terms of the 500 most frequent terms. More specifically, we compute conditional probabilities for a cell in M^{CT} or M^{NT} given a term w (see equation 7.1 and Eq. 7.2). We found several intuitive patterns, which we illustrate below with examples. These are simply presented to motivate the intuition behind the term features used in our predictive model in Section 7.4.

Figure 7.9 shows creation and notification times of task descriptions that contain the terms “church” or “appointment.” The “appointment” plot shows a strong pattern around the morning and midday blocks, representing office hours. Reminders that contain “church” show a clear and intuitive pattern too; they are largely created from Saturday night through Sunday morning, and are set to notify on Sunday early morning and mornings. When we examine the delays between reminder creation and notification, clear patterns emerge.

In Figure 7.10, we compare the average delays of reminders containing the term “appointment” to “laundry.” Clearly, on average, “appointment” reminders have longer delays, reflected by the nature of the task (which may involve other individuals and hence require more planning), whereas “laundry” reminders are more likely to reflect short-term tasks (which may be performed individually).

In summary, we see distinct temporal patterns in task descriptions’ terms. In Section 7.4, we study the generalizability of these patterns.

7.3.5 Temporal Patterns of Notification Times

Finally, we look at correlations between reminders’ creation and notification times. Motivated by the observation that most reminders are set to notify shortly after they are created, we study the probability of a reminder’s notification time given its creation time, $P(r_{NT} | r_{CT})$.

See Figure 7.11 for examples. Looking at the plots in detail, we see how reminders

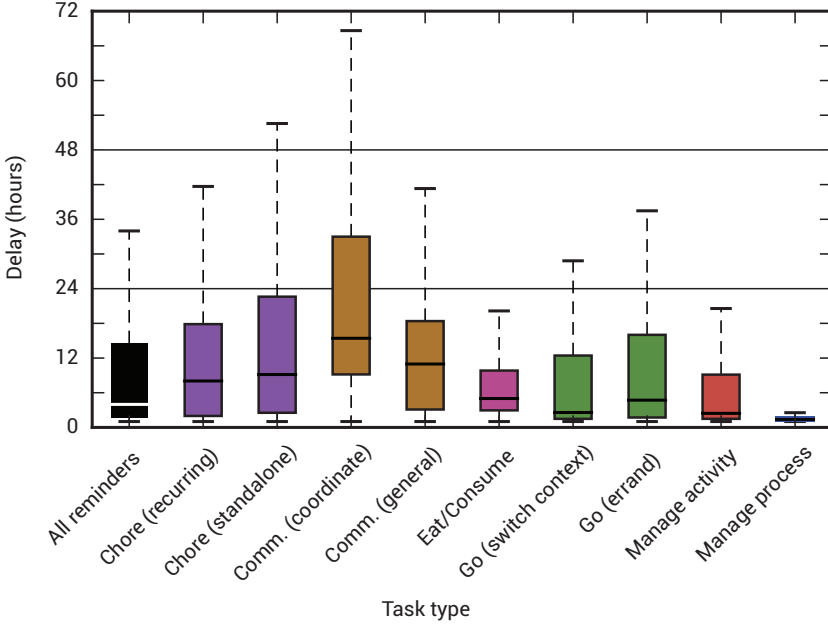


Figure 7.7: Boxplot showing delay between reminder creation and notification times ($n = 125,376$).

across different creation times appear similar: they are most likely to have their notification fire within the same cell or the next, confirming earlier observations that the majority of reminders are short-term (i.e., same cell). However, upon closer inspection, we see that as the reminders’ creation time moves towards later during the day, reminders are more likely to be set to notify the next day. Furthermore, in the third plot from the left, we see how reminders created on Friday evenings have a small but substantial probability of having their notification fire on Monday morning (i.e., the reminder spans the weekend). These patterns show how delay between reminder creation and notification time is low on average, but the length of delay is not independent from the creation time.

In summary, we have shown distinct temporal patterns of reminders of different task types, and of the terms in task descriptions. Finally, we have shown that a reminder’s notification time is most likely set shortly after creation time, but the later in the day a reminder is created, the more likely the notification time is further in the future.

7.4 Predicting Notification Time

In the previous section, we have shown temporal patterns in reminder creation and notification time of four types: aggregate patterns, task type-related, term-based, and time-based. To study whether these patterns can be effectively harnessed, we address a prediction task. Specifically, we turn to the task of predicting the day of the week in which a task is most likely to happen (i.e., predicting r_{NT}). Motivated by our observation that the

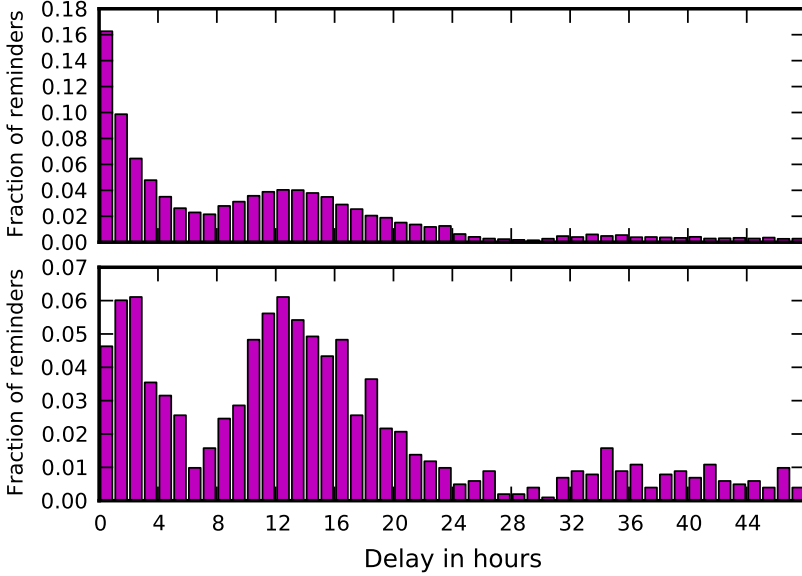


Figure 7.8: Delay (lead time) between reminder creation and notification for “Communicate” subtasks. Showing “Communicate/General” in the top plot, “Communicate/Coordinate” in the bottom.

majority of the reminders are set to trigger soon after being set (Section 7.3.2 and 7.3.5), and by the patterns we observed of the task descriptions’ terms (Section 7.3.4), we aim to answer the following research questions: “*Is the reminder’s creation time indicative of its notification time?*” and “*Do term-based features yield an increase in predictive power?*” The aim of our experiments is not to develop complex or novel predictive models, but instead to study whether the patterns discussed earlier generalize so as to contribute to overall predictive performance.

We cast the task of predicting the day of week a reminder is set to notify as a multiclass classification task, where each day of the week corresponds to a class. The input to our predictive model is the reminder’s task description (r_{task}), creation time (r_{CT}), and the target class is the notification time (r_{NT}) day of week. We measure the predictive power of the patterns identified in the previous sections via term-based and (creation) time-based features. Specifically, for term-based features, we extract bag of word features (unigrams), and our time-based features correspond to R_{CT} ’s time of day (row) and day of week (column), and the minutes since the start of week. See Table 7.9 for an overview.

7.4.1 Experimental Setup

We use Gradient Boosted Decision Trees for classification. This method has proven to be robust and efficient in large-scale learning problems [90]. The ability of this method to deal with non-linearity in the feature space and heterogeneous features makes it a natural choice. To address the multiclass nature of our problem, we employ a one vs. all

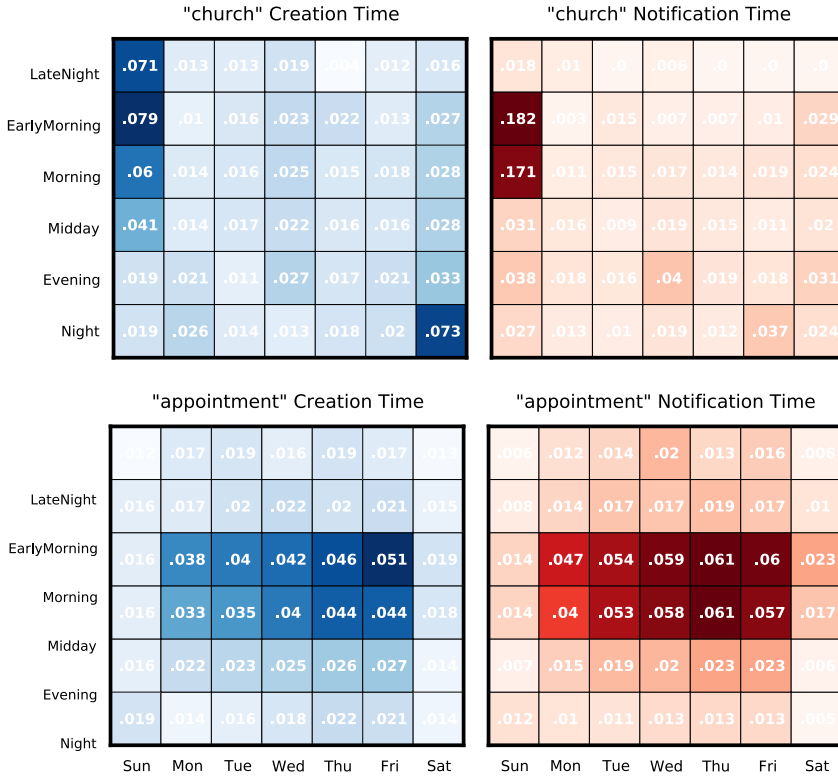


Figure 7.9: Creation and notification times for reminders with the terms “church” (top row) and “appointment” (bottom row).

classification strategy, where we train seven binary classifiers and output the prediction with the highest confidence as final prediction. We compare the accuracy to randomly picking a day of the week (with accuracy of $\frac{1}{7} = 0.1429$) and to a more competitive baseline that predicts the notification will be for the same day that the notification was created (BL-SameDay).

For the experiments, we sample six months of data (January through June 2015). All data were filtered according to the process described in Section 7.1.2, resulting in a total of 1,509,340 reminders. We split this data sequentially: the first 70% (approx. January 1 to May 7) forms the training set and the last 30% (approx. May 8 to June 30) forms the test set. We use the first two months of the training set for the analysis described in Sections 7.1 and 7.3 as well as for parameter tuning before we retrained on the entire training set.

In the next section, we report predictive performance on the held out test set, specifically, we report macro and micro-averaged accuracy over the classes (Macro and Micro, respectively). We compare three approaches: one that leverages time features based on the

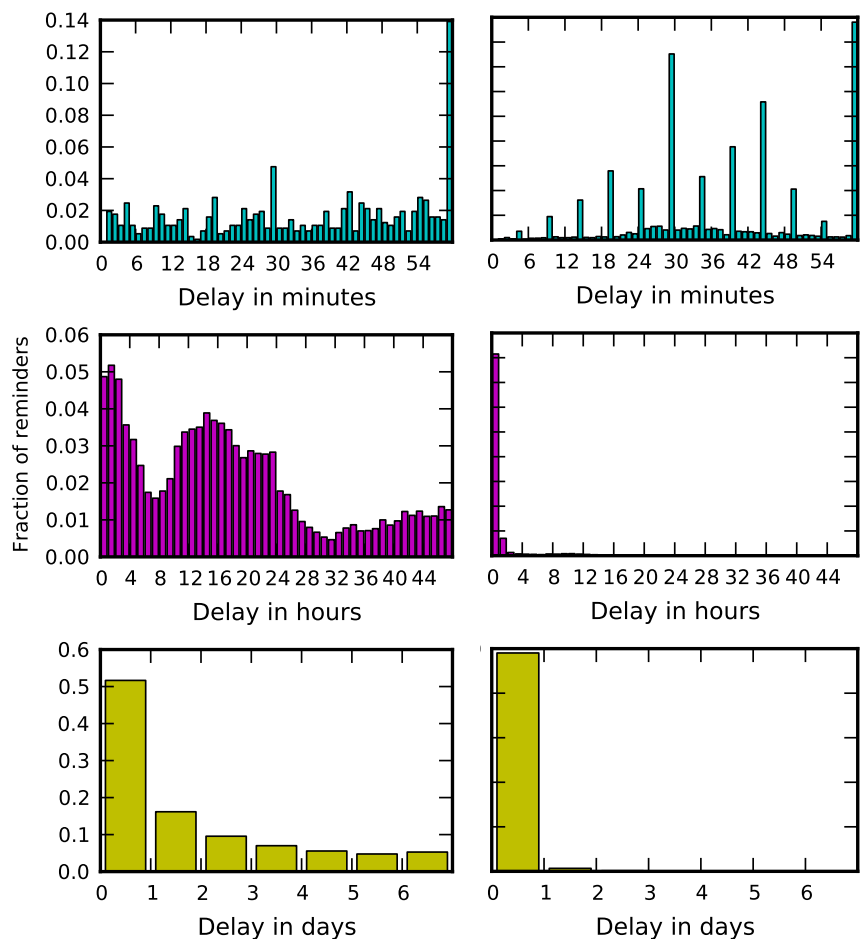


Figure 7.10: Delays between reminder creation and notification for reminder task descriptions containing the terms “appointment” (left column) and “laundry” (right column).

reminder’s creation time (`Time only`), one with term features (`Terms only`), and finally a model that leverages both types of features (`Full model`). We test for statistical significance using t -tests, comparing our predictive models against `BL-SameDay`. The symbols \blacktriangle and \blacktriangledown denote statistically significant differences (greater than the baseline and worse than the baseline, respectively) at $\alpha = 0.01$.

7.4.2 Results

Table 7.10 shows the results of our prediction task. First, we note that the baseline of predicting the notification time to be the same day as the creation time, at 0.5147 micro-averaged accuracy, performs much better than random (at 0.1429). This indicates users

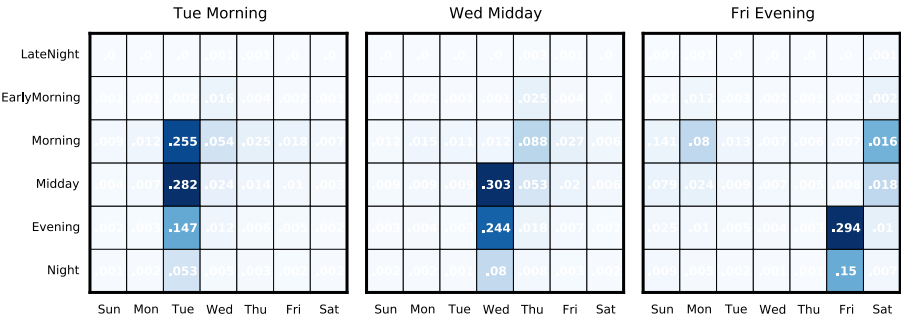


Figure 7.11: Reminder notification time probability distributions over time (i.e., $P(r_{NT} | r_{CT})$), for three different r_{CT} .

Table 7.9: Features used for prediction.

Features	Description
Term Features	Unigram Bag of Word features
Time features	r_{CT} Time of day
	r_{CT} Day of week
	r_{CT} Minutes since start of week

mostly set reminders to plan for events in the short-term. Next, we see that the `Time only` model with a micro-averaged accuracy of 0.6279 significantly improves over the baseline, indicating that the reminder creation time helps further improve prediction accuracy. As noted earlier, tasks planned late at night are more likely to be executed on a different day, and the use of creation time helps leverage this and more general patterns. Finally, the model that uses only features based on the task description (`Terms only`) performs better than random, but does not outperform the baseline. However, when combined with the time model (`Full model`) we see an increase of 8.2% relative to the time only model. We conclude that the creation time provides the most information for predicting when a task will be performed, but the task description provides significant additional information. Framed differently, we show that both the *context* in which digital traces are produced (i.e., the time of day), and the *content* of the digital traces (i.e., the textual description of the task) provide signals for better predicting real-world behavior of the producers of digital traces.

7.5 Conclusion

In this chapter, we have performed a large-scale analysis of reminder data from user activity in a natural setting to answer the following research question:

RQ5 Can we identify patterns in the times at which people create reminders, and, via notification times, when the associated tasks are to be executed?

Table 7.10: Average accuracy of day-of-week prediction task. Statistical significance tested against BL-SameDay.

Run	Micro	Macro	Error reduction
Full model	0.6788 [▲]	0.6761 [▲]	+0.3381
Time only	0.6279 [▲]	0.6258 [▲]	+0.2333
Terms only	0.1777 [▼]	0.1772 [▼]	-0.6944
BL-SameDay	0.5147	0.5165	

We employed a mixed methods approach, comprising data-driven and qualitative methodologies, to extract and identify common task types from reminders that are frequently observed across multiple users. To answer RQ5, we answer several subquestions in this chapter, the findings of which we summarize below.

Task type taxonomy. Through our log analyses we have shown common reminders and made an attempt at identifying and categorizing the types of tasks that underlie them. We identified that the majority of reminders in our sample refer to either daily household chores, running errands, or switching contexts.

Temporal patterns. Next, we show how reminders display different temporal patterns depending on the task type they represent, the reminder’s creation time, and the terms in the task description. Most notably, we have shown that people mostly plan for tasks in the evening hours, and most tasks are set to execute in mornings and throughout the day. On average, people plan in advance for the short-term, i.e., 80% of the reminders in our sample log are set to trigger on the same day as reminder creation.

Prediction task. Finally, we demonstrated that we can leverage these patterns to predict the day of the week that a reminder is most likely to trigger, i.e., the day the task is most likely to be executed. Specifically, we confirm that the reminder’s creation time is a strong indicator of notification time, but that including the task description further improves accuracy over the strongest baseline, with a 33% reduction in error.

In line with our findings in the previous chapter, here we confirm that both the *context* in which digital traces are produced (the time of day), and the (textual) *content* of the digital traces (the reminder description) provide valuable signals in predicting the real-world behavior and activity of our entities of interest: the producers of digital traces. Furthermore, the findings in this chapter have implications for designing systems to help with task completion, and more generally for developing technology to reduce prospective memory failures. The analysis and prediction task show that we can leverage large-scale interaction logs to predict a users’ (planned) activities.

There are several limitations in our log analyses. First, we performed this analysis on a specific subset of reminders: reminders from one geographic locale and for a single type of reminder: time-based. There are opportunities to understand cultural and linguistic factors in reminder creation by considering reminders from multiple regions. We additionally seek

to investigate other types of reminders, such as those involving people, places, and events. Second, it is difficult to quantify the comprehensiveness of the task type taxonomy, which covers common reminders. The ontology may therefore not cover more intricate, personal, specific, or complex reminders, the nature of which needs to be better understood. Finally, our approach and analysis is entirely log based. The taxonomy's categories were manually labeled, and we make inferences and assumptions about the tasks that people are engaged in. User studies are needed to better understand the reminder process, including the generation and value of reminders, including how people behave when they are notified.

Future work includes developing more sophisticated models (e.g., considering personalized signals) to improve prediction performance. Besides the value of the insights gained in this chapter from the discovery perspective, i.e., from understanding the link between people's interaction logs with intelligent assistants and their (planned) activities and tasks, the insights and predictions about tasks and the use of reminders can also prove valuable with the development of systems endowed with the ability to proactively reserve time, manage conflicts, remind people about tasks they might forget, and, more generally, help people achieve their goals.

8

Conclusions

“Si finis bonus est, totum bonum erit.”

—*Gesta Romanorum*, Tale LXVII

In this thesis, we have studied *discovery* in digital traces. We have done so in two parts: in Part I we have studied the *content* of (textual) digital traces, and in Part II we have turned to the *contexts* in which digital traces are created, with the goal of leveraging digital traces for predicting real-world activity.

Part I revolves around semantic search. Semantic search is a search paradigm that uses structured knowledge to increase retrieval effectiveness [250], and allows us to better support the exploratory search process that is inherent to the discovery process [268]. More specifically, in this part our entities of interest are the real-world entities that occur (i.e., are mentioned) in digital traces. We leverage publicly available knowledge bases such as Wikipedia and Freebase, and study *emerging entities*: entities that are not (yet) described in the knowledge base. This is motivated by the fact that in typical discovery scenarios (e.g., investigative journalism or digital forensics), the entities of interest may not be known a priori. In three chapters, we analyze the temporal patterns of emerging entities in online text streams (Chapter 3), predict them in social media streams (Chapter 4), and enrich their representations by leveraging “collective intelligence” (i.e., how people refer to and search for the entities on the web), for increased retrieval effectiveness (Chapter 5).

The second part of this thesis revolves around the contexts in which digital traces are created. Here, we shift our focus, and our entities of interest are the producers of digital traces, i.e., the people who leave behind digital traces. Our goal is to predict real-world activity from digital traces, a core task in E-Discovery [197]. We present two case studies: in the first, we study enterprise email collections, and we analyze aspects that govern communication between employees and aim to predict email communication by leveraging the communication graph and email content (Chapter 6). Next, we study user interaction logs with a personal intelligent assistant (Microsoft Cortana), that represents the users’ planned activities and tasks (Chapter 7). In this chapter, we show common usage patterns of the reminder service, and show how we can leverage the uncovered patterns for predicting the most likely day at which a task will be executed.

In this final chapter of the thesis, we re-iterate the research questions we answered, and summarize our methods and findings in Section 8.1. In Section 8.2 we reflect on future work.

8.1 Main Findings, Limitations, and Implications

Here, we summarize the methods, findings, and their implications. We do so by answering the research questions raised in Chapter 1.

Part I. Analyzing, Predicting, and Retrieving Emerging Entities

In the first content chapter of this thesis (Chapter 3), we start our exploration of real-world entities in digital traces with a large-scale analysis of how entities emerge. In Chapter 3 we answer our first research question:

RQ1 Are there common temporal patterns in how entities of interest emerge in online text streams?

To answer this question, we analyze a large collection of entity time series, i.e., time series of entity mentions in the lead time between an entity’s first mention in online text streams and its subsequent incorporation into the knowledge base.

We apply an unsupervised hierarchical clustering method to uncover groups of entities that exhibit different emergence patterns. We discover two main emergence patterns. First, roughly half of the entities we study emerge in a “bursty” fashion, i.e., they surface in online text streams without a precedent, and blast into activity before they are incorporated in the KB. Other entities display a “delayed” pattern, where they appear in public discourse, experience a period of inactivity, and then resurface before being incorporated into the KB. In addition to these emergence patterns, we found distinct differences in the emergence patterns of different types of entities, suggesting that different entities of interest exhibit different emergence patterns.

The work presented in this chapter has several implications. Most notably, the analyses have implications for designing systems to detect emerging entities before they are part of the KB. We have shown how entities tend to resurface multiple times before being added to Wikipedia. This suggests that burst detection may be an effective way of “catching” emerging entity before they are part of the KB. Furthermore, we have shown that the different streams in which the entities emerge are indicative of how fast the entity may be added to the KB.

The work presented in this chapter also know several limitations. First, evaluating unsupervised clustering without labels is not trivial: Von Luxburg et al. [254] aptly question whether clustering is an “art or science.” However, we do have indications to believe the clusters we generated are meaningful. First, the cluster signatures show distinct and statistically significantly different patterns, whereas different groupings of time series (e.g., by stream or by entity types) did not yield any discernible patterns. Furthermore, the structure of the dendrogram (i.e., the hierarchical cluster tree) suggests there are clear subgroups in the data.

Another limitation to our study is related to the dataset used. First, we rely on automatically generated annotations (FAKBA1 dataset), which means we cannot assume a 100% accurate set of annotations. What’s more, the set of long-tail entities that are likely in our subset of “emerging entities,” are known to be comparatively more difficult to link, as most entity linking methods have a strong reliance on “popularity”-based features, such as, commonness [181]. Next to the quality of the annotations, the coverage or selection of

sources in the data may be a concern for how well our findings generalize to other domains or datasets. Sampling bias, and e.g., the absence of popular social media platforms (such as Facebook, Twitter, or Tumblr) means the findings may be different with other datasets. Finally, the cultural bias that is inherent to the dataset selection. By studying only English sources, we effectively studied how entities emerge in the English speaking world.

The analysis of emerging entities in Chapter 3 shows that knowledge bases are never complete: new entities may emerge as events unfold, but at the same time, long-tail, relatively unknown entities may also be added to a KB. As a follow-up to this chapter, we turn our attention to predicting newly emerging entities in Chapter 4. More specifically, we focus on predicting mentions of emerging KB entities in social media streams, motivated by the high pace and noisy nature that characterizes social media streams. Our method leverages an entity linking system to automatically generate training data for a named-entity recognizer to learn to recognize KB entities. We answer:

RQ2 Can we leverage prior knowledge of entities of interest to bootstrap the discovery of new entities of interest?

To answer this question, we propose a novel unsupervised method for generating pseudo-training data to train a named-entity recognizer and classifier (NERC) for predicting mentions of emerging entities that are likely to be added to a KB. The method includes two sampling methods for selecting high quality training samples, the first selects samples based on the textual quality of the sample, and the second leverages the confidence score of the entity linking system to select samples of which the entity linker is more confident, in an attempt to reduce noisy samples. We measure the effectiveness of these sampling methods by their impact on the accuracy of predicting newly emerging entities. Furthermore, we perform an additional analysis, where we study the impact of the amount of prior knowledge. We approximate this quantity of prior knowledge by randomly sampling different (size) reference knowledge bases, i.e., we leave out a part of the knowledge base, and study the prediction effectiveness.

We find that sampling by textual quality improves the performance of our NERC method and consequently our method’s performance in predicting emerging entities. Furthermore, we show that setting a higher threshold on the entity linking system’s confidence score for generating pseudo-ground truth results in fewer labels but better performance. We show that the NERC is better able to separate noise from entities that are worth including in a knowledge base. Finally, we show that in the case of a small amount of prior knowledge, i.e., limited size of the available initial knowledge, our method is able to cope with missing labels and incomplete data, as observed through its consistent and stable precision. This finding justifies our proposed method that assumes incomplete data by design.

The implication of our findings is that we our approach can effectively support the exploratory search process, by retrieving similar entities to those in a reference KB, that are not yet part of the KB.

A limitation of our work is the retrospective scenario we employ in our evaluation. For this reason, we cannot measure the impact of the novelty of entities of interest as they emerge.

Finally, in Chapter 5 we address the follow-up task of enriching entity representations of emerging entities with descriptions coming from a variety of sources, in a real-time, streaming manner. Collecting entity descriptions from different sources, and combining them in a single representation for improved retrieval effectiveness, is beneficial in scenarios where information about the same entities is spread over multiple sources, e.g., in a discovery scenario, where multiple sources that represent the activity of, e.g., people in a company (e.g., email collections, collaboration platforms, social media posts) can be used to paint the full picture. We answer:

RQ3 Can we leverage collective intelligence to construct entity representations for increased retrieval effectiveness of entities of interest?

We answer this question by proposing an effective way of leveraging collective intelligence (i.e., entity descriptions from multiple sources) to construct entity representations for optimal retrieval effectiveness of entities of interest. More specifically, we collect descriptions from knowledge bases, social media, and the web, to increase retrieval effectiveness of entities. We do this in a dynamic scenario, i.e., we distinguish between static description sources, where descriptions are aggregated and added to entity representations, and dynamic description sources, where descriptions come in a real-time, streaming manner.

The main challenge with leveraging collective intelligence is the heterogeneity between entities: some entities may receive many descriptions (i.e., head entities), while others may receive few (tail entities). At the same time, different description sources represent very different content, compare, e.g., the structured and clean entity description from a knowledge base, to the contexts in which people refer to entities on social media.

First, we demonstrate that incorporating dynamic description sources into dynamic collective entity representations enables a better matching of users' queries to entities, resulting in an increase of entity ranking effectiveness. In addition, we show that informing the ranker on the expansion state of the entities, further increases the ranking effectiveness. Finally, we show how retraining the ranker leads to improved ranking effectiveness in dynamic collective entity representations, however, even a static (not periodically retrained) ranker's performance improves over time, suggesting that even the static ranker benefits from newly incoming descriptions.

The findings in this chapter imply that information from different types of sources can be effectively combined to improve retrieval effectiveness of entities of interest. The scenario of different (heterogeneous) data sources that may be related, is not uncommon in the E-Discovery scenario, e.g., when an email box and associated attachments or files can all be traced to a single entity.

One limitation in the work presented in this chapter was related to the experimental design, which was constrained by the lack of temporally aligned and sizeable datasets. In particular, the temporal misalignment between different corpora prevents the analysis of temporal patterns that affect entities in unforeseen ways.

Part II. Analyzing and Predicting Activity from Digital Traces

In the first chapter of this part, the digital traces under study are enterprise email, and the entities of interest are emailers.

Our aim in Chapter 6 is to provide insights in the aspects that guide communication between people. Specifically, we study the impact of aspects of the communication graph (e.g., the strength of ties between emailers, or the proximity of emailers in the enterprise network), and the impact of email content (e.g., the similarity of email content between two emailers). We apply these signals for predicting enterprise email communication, which may find application in detecting unexpected communication. We answer:

RQ4 Can we predict email communication through modeling email content and communication graph properties?

We present a hybrid model for email recipient prediction that leverages both the information from the email network’s communication graph, and the Language Models of the emailers, estimated with the content of the emails that are sent by each user. Our model starts from scratch, in that it does not assume or need seed recipients, and it is updated for each email sent.

We show that the communication graph properties and the email content signals both provide a strong baseline for predicting recipients, but are complementary, i.e., combining both signals achieves the highest accuracy for predicting the recipients of email. Furthermore, we show that the number of received emails is an effective method for estimating the prior probability of observing a recipient, and we show that the number of emails sent between two users is an effective way of estimating the “connectedness” between the two users.

The implication of our findings is that both the *context* in which digital traces are created (i.e., the position of an emailer in the communication graph, her ties with surrounding emailers), and the *content* (i.e., the email itself) are important signals in predicting real-world behavior (i.e., email communication).

Next, we study interaction logs with an intelligent personal assistant in Chapter 7. Intelligent assistants are proliferating across (both mobile and desktop) devices, they have a close proximity to and embedding in our day-to-day life (in part thanks to their conversational and “personal” nature). For these reasons, interaction logs with personal assistants are a rich resource for digital evidence. In this chapter we take a similar approach to the previous, and analyze a novel dataset of interaction logs, and address a prediction task to study how digital traces can be employed to predict real-world activity of our entities of interest.

User interaction logs of intelligent personal assistants may contain many rich contextual clues, and represent the user’s (planned) activities or geographic location at any given time. We are particularly interested in the reminder service of one such personal assistant (Microsoft Cortana), as this service is a heavily used feature by its users, and reminders represent (planned) tasks of users, and can hence be used to gain insights into their (planned) activities. In combination with other data, these logs could be a rich signal for inferring real-world activity, a core task in discovery. We answer:

RQ5 Can we identify patterns in the times at which people create reminders, and, via notification times, when the associated tasks are to be executed?

We analyze a large-scale user log, comprising over 500,000 time-based reminders of over 90,000 users. We identify a body of common tasks types that give rise to the reminders

across a large number of users. We subsequently arrange these tasks into a taxonomy. Finally, we study their temporal patterns, and we address a prediction task, where we aim to predict when a task reminder is set to trigger, i.e., the user aims to execute the task, given the reminder task description and creation time.

We show how reminders display different temporal patterns depending on the task type they represent, the reminder’s creation time, and the terms in the task description. We show that the time at which a user creates a reminder is a strong indication of when the task is scheduled to be executed. Furthermore, we show that including the text of the task reminder further improves prediction accuracy. Much like in Chapter 6, we confirm that combining the *content* of the digital traces (i.e., the textual description of the reminder), and the *context* in which they are created (i.e., the time of day), achieves the highest prediction accuracy, suggesting that both signals are complimentary.

As in the previous chapter, the findings in this chapter imply that both the *context* (the time of day), and the *content* (the reminder description) of digital traces provide signals that help in predicting real-world behavior of our entities of interest: the producers of digital traces.

This chapter is not without limitations either. First, we make assumptions about the behavior and activities of people, based on interaction logs. Without conducting user studies or using other ways to infer the actual activities of people — beyond what they say to Cortana, we cannot make any statements about people’s activities.

8.2 Future Research Directions

In this section, we summarize some of the limitations to the work presented in this thesis, and identify some areas for potential follow-up work.

Part I. Analyzing, Predicting, and Retrieving Emerging Entities

Chapter 3 is a starting point of studying how entities emerge in public discourse, and what happens in the lead time between the entity first surfaces, and is subsequently deemed “important enough” to be added to the KB. As a next step, we should take a closer look at the detailed circumstances under which entities emerge, by not only considering the number of documents they appear in over time, but also in which contexts, e.g., by looking at the content of the articles themselves. Another interesting aspect of emerging entities, that falls out of the scope of the present work, is the notion of when entities are “deemed important enough,” i.e., when the collective reaches consensus. For example, one could study the emergence patterns of entities that are removed from the KB. Finally, the observations made in this chapter could be explored in a prediction task, where, e.g., given a partial entity time series, the task would be to predict the point at which the entity will be incorporated in the KB.

The method we present in Chapter 4 addresses entity mention detection for emerging entities, i.e., we leverage a knowledge base to label entity mentions, so a named-entity recognizer can identify similar but unlinked mentions in social media posts. A natural follow-up to would be to “close the loop,” i.e., feed back the emerging entities to populate the knowledge base. Closing the loop would open the door to a fully automated knowledge

base population process, the scenario where the knowledge base is automatically populated with entity mentions and representations, which in turn means the pseudo-training data generated by the entity linking system remains up-to-date at all times.

To achieve this, an additional step of constructing seed entity representations is necessary. One approach is to mine keyphrases [120] from the tweets in which the emerging entities appear. To effectively construct these representations, an entity clustering or disambiguation step would be beneficial, to collect enough content to extract keyphrases from [55, 215].

Another challenge that lies in this direction is that of supervision; letting the system roam free may introduce noise (i.e., false positives) into the knowledge base, which may over time corrupt training data and hence derail the system. Studying ways of incorporating a “human-in-the-loop,” when evaluating and judging predictions, using e.g., crowdsourcing finds applications in any task in which a self-learning, automated, system runs freely [60, 84]. The human-in-the-loop paradigm for evaluating and tracking algorithmic predictions is of particular interest in the E-Discovery domain, where machine-generated predictions can have severe impact on the outcomes of, e.g., legal cases [197].

A natural extension to the work presented in Chapter 5 is to study our method in a real life scenario, with real-time, streaming data, and to provide a more fine-grained analysis of the impact of the importance features and to study the link between the temporal patterns of entities (as studied in Chapter 3) and their retrieval effectiveness.

Larger and temporally aligned data collections would also increase the potential challenge of “swamping” [223] or “document vector saturation” [138], where description sources may completely overtake an entity representation. This swamping phenomenon could be addressed by incorporating a temporal decay on the terms that make up the representation, or imposing a size limit on the fields. Furthermore, additional feature engineering, e.g. informing the ranker of the diversity of terms in a field, or the novelty of the terms compared to the original entity representation, may also prove beneficial when the number and volume of descriptions increase.

In addition, we focused on a concrete end-to-end task in this chapter, we evaluate the quality of the representations by measuring the retrieval effectiveness, given the added entity descriptions. Another direction would be to study the shifting entity representations from a more analytic perspective. One could study the entity drift, or whether entity representations change over time in meaningful ways, e.g., by studying which words are more strongly associated to entities at different points in time. Similar to studying how word use may shift over time [139], studying entity representation drift may provide insights into the type of entities or events that are prone to change more or less intensely.

Part II. Analyzing and Predicting Activity from Digital Traces

Our study in Chapter 6 focused on the (relative) contribution of two different aspects; communication graph and email content properties. Engineering-wise, however, prediction accuracy could be improved by a number of approaches: first, employing the scores of our generative model as features in a machine learning model, instead of using them directly for ranking, means multiple signals could be easily combined, and prediction accuracy may increase. Next, taking a more fine-grained or local approach to the communication graph could further boost prediction accuracy, by incorporating (implicit) organizational

structures of the enterprise. Finally, the analysis of our experimental results highlighted that dealing with growing networks, and handling time in general, proved a challenge. Incorporating time-awareness into both components of the model may prove beneficial for performance over time, and studying the effect of communication patterns in evolving networks [3, 248], may alleviate these weaknesses, and further improve our model's prediction accuracy.

Another opportunity for the work presented in this chapter is to adapt the presented method to novel tasks. One could, e.g., look at role detection or inferring a company's hierarchical structure through leveraging both social network aspects and email content (e.g., common topics amongst employees). Furthermore, a predictive model with accurate recipient predictions could be employed for rating the "likelihood" of historic interactions, and thus be applied for outlier or anomaly detection, to discover communication in an enterprise that is "unexpected," and therefore may be of interest to, e.g., a digital forensic analyst.

The prediction task studied in Chapter 7 served as a starting point for further research into predicting (future) tasks and activities given historic signals. Future work includes developing more sophisticated models. Whereas the current work focused on global patterns, seen frequently and across many users, considering more personalized signals may increase prediction accuracy; it is likely that users exhibit distinct patterns and usage of the reminder service. Furthermore, incorporating more signals into the task, e.g., leveraging geographical signals and location services to better infer the current context of the user, to be able to distinguish between being at work, or at home, or driving/in transit, or even a more fine-grained context, by looking up the type of location the user currently is at (e.g., a restaurant, grocery store).

Furthermore, more sophisticated approaches to understanding the language of the reminder descriptions may prove beneficial in clustering similar task reminders, to battle sparsity and yield stronger (temporal) signals for different task types. Using topic modeling, distributional or dimension reduction techniques (through e.g., word or reminder embeddings) could prove beneficial in clustering similar task descriptions.

Finally, the insights and predictions about tasks and the use of a personal assistant's reminder service could also prove valuable in a more practical or commercial setting, e.g., with developing systems endowed with the ability to proactively reserve time for users, automatically manage their scheduling and planning conflicts, remind people about tasks they might forget, and, more generally, to help people achieve their goals.

- [1] Oxford dictionary (<http://www.oxforddictionaries.com/definition/english/entity>), 2016. [Online; accessed 13-September-2016]. (Cited on page 3.)
- [2] D. Aberdeen, O. Pacovsky, and A. Slater. The learning behind gmail priority inbox. NIPS 2010 Workshop on Learning on Cores, Clusters and Clouds, 2010. (Cited on page 25.)
- [3] C. Aggarwal and K. Subbian. Evolutionary network analysis: A survey. *ACM Computing Surveys*, 47: 10:1–10:36, 2014. (Cited on page 146.)
- [4] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. SIGIR, pages 19–26. ACM, 2006. (Cited on pages 7 and 24.)
- [5] L. Akoglu, M. McGlohon, and C. Faloutsos. Oddball: Spotting anomalies in weighted graphs. PAKDD, pages 410–421. Springer, 2010. (Cited on page 109.)
- [6] E. Amitay, A. Darlow, D. Konopnicki, and U. Weiss. Queries as anchors: selection by association. HYPERTEXT, pages 193–201. ACM, 2005. (Cited on page 23.)
- [7] M. G. Armentano and A. A. Amandi. Recognition of user intentions for interface agents with variable order markov models. UMAP, pages 173–184. Springer-Verlag, 2009. (Cited on page 25.)
- [8] R. Baeza-Yates, C. Hurtado, M. Mendoza, and G. Dupret. Modeling user search behavior. Latin American Web Congress, page 242. IEEE, 2005. (Cited on page 7.)
- [9] N. Balasubramanian and S. Cucerzan. Topic pages: An alternative to the ten blue links. ICSC, pages 353–360. IEEE, 2010. (Cited on page 17.)
- [10] K. Balog and M. de Rijke. Finding experts and their details in e-mail corpora. WWW, pages 1035–1036. ACM, 2006. (Cited on page 25.)
- [11] K. Balog and K. Nørkvåg. On the use of semantic knowledge bases for temporally-aware entity retrieval. ESAIR, pages 1–2. ACM, 2012. (Cited on page 23.)
- [12] K. Balog, M. Bron, and M. de Rijke. Category-based query modeling for entity search. ECIR, pages 319–331. Springer-Verlag, 2010. (Cited on pages 23 and 81.)
- [13] K. Balog, A. de Vries, P. Serdyukov, P. Thomas, and T. Westerveld. Overview of the TREC 2009 entity track. TREC. NIST, 2010. (Cited on pages 22, 77, and 89.)
- [14] K. Balog, P. Serdyukov, and A. de Vries. Overview of the TREC 2010 entity track. TREC. NIST, 2011. (Cited on pages 22, 77, and 89.)
- [15] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su. Optimizing web search using social annotations. WWW, pages 501–510. ACM, 2007. (Cited on page 23.)
- [16] K. Barmapsalou, D. Damopoulos, G. Kambourakis, and V. Katos. A critical review of 7 years of mobile device forensics. *Digital Investigation*, 10:323–349, 2013. (Cited on page 115.)
- [17] D. Barreau and B. A. Nardi. Finding and reminding: File organization from the desktop. *SIGCHI Bull.*, 27:39–43, 1995. (Cited on page 26.)
- [18] D. Barua, J. Kay, B. Kummerfeld, and C. Paris. Modelling long term goals. UMAP, pages 1–12. Springer, 2014. (Cited on page 25.)
- [19] A. E. C. Basave, A. Varga, M. Rowe, M. Stankovic, and A.-S. Dadzie. Making sense of microposts (#msm2013) concept extraction challenge. MSM, pages 1–15. CEUR-WS, 2013. (Cited on page 67.)
- [20] H. Bast, B. Buchhold, and E. Haussmann. Semantic search on text and knowledge bases. *Foundations and Trends in Information Retrieval*, 10:119–271, 2016. (Cited on page 4.)
- [21] M. Becker, B. Hachey, B. Alex, and C. Grover. Optimising selective sampling for bootstrapping named entity recognition. ICML Workshop on Learning with Multiple Views, pages 5–11, 2005. (Cited on page 22.)
- [22] R. Bekkerman, A. McCallum, and G. Huang. Automatic categorization of email into folders: Benchmark experiments on Enron and SRI corpora. *Center for Intelligent Information Retrieval, Technical Report IR*, 418, 2004. (Cited on pages 25 and 90.)
- [23] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. AAAIWS, pages 359–370. AAAI Press, 1994. (Cited on page 39.)
- [24] B. Billerbeck, G. Demartini, C. Firan, T. Iofciu, and R. Krestel. Ranking entities using web search query logs. ECDL, pages 273–281. Springer, 2010. (Cited on page 23.)
- [25] C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *International Journal on Semantic Web and Information Systems*, 5:1–22, 2009. (Cited on page 17.)
- [26] K. Bontcheva and D. Rout. Making sense of social media streams through semantics: a survey. *Semantic Web journal*, 5:373–40, 2014. (Cited on pages 4 and 17.)
- [27] K. Bontcheva, L. Derczynski, A. Funk, M. A. Greenwood, D. Maynard, and N. Aswani. TwitIE: An open-source information extraction pipeline for microblog text. RANLP, pages 83–90. Association for

8. Bibliography

- Computational Linguistics, 2013. (Cited on page 20.)
- [28] A. Bordes, J. Weston, R. Collobert, and Y. Bengio. Learning structured embeddings of knowledge bases. AAAI, pages 301–306. AAAI Press, 2011. (Cited on page 24.)
- [29] E. Boschee, R. Weischedel, and A. Zamanian. Automatic information extraction. In *International Conference on Intelligence Analysis*, 2005. (Cited on page 35.)
- [30] M. Brandimonte, G. O. Einstein, and M. A. McDaniel, editors. *Prospective Memory: Theory And Applications*. Psychology Press (Taylor & Francis Group), 1996. (Cited on page 26.)
- [31] A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002. (Cited on page 13.)
- [32] A. Broder, E. Gabrilovich, V. Josifovski, G. Mavromatis, D. Metzler, and J. Wang. Exploiting site-level information to improve web search. CIKM, pages 1393–1396. ACM, 2010. (Cited on page 24.)
- [33] M. Bron, B. Huurnink, and M. de Rijke. Linking archives using document enrichment and term selection. TPD, pages 360–371. Springer, 2011. (Cited on page 19.)
- [34] L. Buitinck and M. Marx. Two-stage named-entity recognition using averaged perceptrons. NLDB, pages 171–176. Springer-Verlag, 2012. (Cited on page 66.)
- [35] R. Bunescu. Using encyclopedic knowledge for named entity disambiguation. EACL, pages 9–16. Association for Computational Linguistics, 2006. (Cited on page 21.)
- [36] D. Carmel, M.-W. Chang, E. Gabrilovich, B.-J. P. Hsu, and K. Wang. ERD 2014: Entity recognition and disambiguation challenge. *SIGIR Forum*, 2014. (Cited on page 22.)
- [37] D. Carmel, G. Halawi, L. Lewin-Eytan, Y. Maarek, and A. Raviv. Rank by time or by relevance?: Revisiting email search. CIKM, pages 283–292. ACM, 2015. (Cited on page 14.)
- [38] V. R. Carvalho and W. W. Cohen. On the collective classification of email “speech acts”. SIGIR, pages 345–352. ACM, 2005. (Cited on page 25.)
- [39] V. R. Carvalho and W. W. Cohen. Ranking users for intelligent message addressing. ECIR, pages 321–333. Springer, 2008. (Cited on page 25.)
- [40] E. Casey. *Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet with Cdrom*. Academic Press, Inc., 2011. (Cited on page 115.)
- [41] T. Cassidy, Z. Chen, J. Artilles, H. Ji, H. Deng, L. Ratnov, J. Zheng, J. Han, and D. Roth. CUNY-UIUC-SRI TAC-KBP2011 entity linking system description. TAC. NIST, 2011. (Cited on page 19.)
- [42] T. Cassidy, H. Ji, L.-A. Ratnov, A. Zubiaga, and H. Huang. Analysis and enhancement of wikification for microblogs with context expansion. COLING, pages 441–456. Association for Computational Linguistics, 2012. (Cited on pages 19 and 20.)
- [43] P. Chaurasia, S. McClean, C. D. Nugent, and B. Scotney. A duration-based online reminder system. *International Journal of Pervasive Computing and Communications*, 10(3):337–366, 2014. (Cited on page 26.)
- [44] W. W. Cohen, V. R. Carvalho, and T. M. Mitchell. Learning to classify email into “speech acts”. EMNLP, pages 309–316. Association for Computational Linguistics, 2004. (Cited on page 25.)
- [45] M. Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. EMNLP, pages 1–8. Association for Computational Linguistics, 2002. (Cited on page 66.)
- [46] G. V. Cormack, M. R. Grossman, B. Hedin, and D. W. Oard. Overview of the TREC 2010 legal track. TREC. NIST, 2011. (Cited on pages 25 and 104.)
- [47] S. Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. EMNLP, pages 708–716. Association for Computational Linguistics, 2007. (Cited on pages 19 and 20.)
- [48] S. Cucerzan. TAC entity linking by performing full-document entity extraction and disambiguation. TAC. NIST, 2011. (Cited on page 19.)
- [49] A. Culotta, R. Bekkerman, and A. McCallum. Extracting social networks and contact information from email and the web. Conference on Email and Anti-Spam (CEAS), 2004. (Cited on page 25.)
- [50] M. Czerwinski, E. Horvitz, and S. Wilite. A diary study of task switching and interruptions. CHI, pages 175–182. ACM, 2004. (Cited on page 26.)
- [51] L. A. Dabbish, R. E. Kraut, S. Fussell, and S. Kiesler. Understanding email use: predicting action on a message. CHI, pages 691–700. ACM, 2005. (Cited on page 25.)
- [52] J. Dalton, L. Dietz, and J. Allan. Entity query feature expansion using knowledge base links. SIGIR, pages 365–374. ACM, 2014. (Cited on page 17.)
- [53] J. Dalton, J. R. Frank, E. Gabrilovich, M. Ringgaard, and A. Subramanya. Fakba1: Freebase annotation of trec kba stream corpus, version 1 (release date 2015-01-26, format version 1, correction level 0), 2015. (Cited on pages 35 and 36.)
- [54] A. S. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: Scalable online collaborative filtering. WWW, pages 271–280. ACM, 2007. (Cited on page 2.)

-
- [55] A. Davis, A. Veloso, A. S. da Silva, W. Meira, Jr., and A. H. F. Laender. Named entity disambiguation in streaming data. *ACL*, pages 815–824. Association for Computational Linguistics, 2012. (Cited on page 145.)
 - [56] M. De Choudhury, M. Gamon, S. Counts, and E. Horvitz. Predicting depression via social media. *ICWSM. AAAI*, 2013. (Cited on page 2.)
 - [57] A. de Vries, A.-M. Vercoustre, J. A. Thom, N. Craswell, and M. Lalmas, editors. *Overview of the INEX 2007 entity ranking track*, INEX, 2008. Springer. (Cited on pages 22, 77, and 89.)
 - [58] G. Demartini, A. de Vries, T. Iofciu, and J. Zhu, editors. *Overview of the INEX 2008 entity ranking track*, INEX, 2009. Springer-Verlag.
 - [59] G. Demartini, T. Iofciu, and A. de Vries, editors. *Overview of the INEX 2009 entity ranking track*, INEX, 2010. Springer-Verlag. (Cited on pages 22, 77, and 89.)
 - [60] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux. Zencrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. *WWW*, pages 469–478. ACM, 2012. (Cited on page 145.)
 - [61] L. Derczynski, D. Maynard, N. Aswani, and K. Bontcheva. Microblog-genre noise and impact on semantic annotation accuracy. *HYPERTEXT*, pages 21–30. Association for Computational Linguistics, 2013. (Cited on page 20.)
 - [62] L. Derczynski, D. Maynard, G. Rizzo, M. van Erp, G. Gorrell, R. Troncy, J. Petrak, and K. Bontcheva. Analysis of named entity recognition and linking for tweets. *Information Processing & Management*, 51: 32–49, 2015. (Cited on page 20.)
 - [63] R. W. DeVaul, B. Clarkson, and A. S. Pentland. The memory glasses: Towards a wearable, context aware, situation-appropriate reminder system. In *Proceedings of the SIGCHI Workshop on Situated Interaction in Ubiquitous Computing*, 2000. (Cited on pages 25 and 26.)
 - [64] A. K. Dey and G. D. Abowd. Cybreminder: A context-aware system for supporting reminders. *HUC*, pages 172–186, London, UK, UK, 2000. Springer-Verlag. (Cited on page 25.)
 - [65] D. Di Castro, Z. Karnin, L. Lewin-Eytan, and Y. Maarek. You’ve got mail, and here is what you could do with it!: Analyzing and predicting actions on email messages. *WSDM*, pages 307–316. ACM, 2016. (Cited on page 25.)
 - [66] C. P. Diehl, L. Getoor, and G. Namata. Name reference resolution in organizational email archives. *SDM*, pages 20–22. SIAM, 2006. (Cited on page 25.)
 - [67] J. Diesner, T. L. Frantz, and K. M. Carley. Communication networks from the enron email corpus “it’s always about the people. enron is no different”. *Computational & Mathematical Organization Theory*, 11:201–228, 2005. (Cited on page 25.)
 - [68] R. K. Dismukes. Prospective memory in workplace and everyday situations. *Current Directions in Psychological Science*, 21(4):215–220, 2012. (Cited on page 26.)
 - [69] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmman, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. *KDD*, pages 601–610. ACM, 2014. (Cited on page 20.)
 - [70] S. Dumais, E. Cutrell, J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins. Stuff i’ve seen: A system for personal information retrieval and re-use. *SIGIR*, pages 72–79. ACM, 2003. (Cited on page 14.)
 - [71] M. Efron, P. Organisciak, and K. Fenlon. Improving retrieval of short texts through document expansion. *SIGIR*, pages 911–920, 2012. (Cited on page 23.)
 - [72] G. O. Einstein and M. A. McDaniel. Normal aging and prospective memory. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 16(4):717–726, 1990. (Cited on page 26.)
 - [73] N. Eiron and K. S. McCurley. Analysis of anchor text for web search. *SIGIR*, pages 459–460. ACM, 2003. (Cited on pages 23 and 79.)
 - [74] J. Ellis and L. Kvavilashvili. Prospective memory in 2000: Past, present, and future directions. *Applied Cognitive Psychology*, 14(7):S1–S9, 2000. (Cited on page 26.)
 - [75] T. Elsayed and D. W. Oard. Modeling identity in archival collections of email: A preliminary study. *Conference on Email and Anti-Spam (CEAS)*, pages 95–103, 2006. (Cited on page 25.)
 - [76] T. Elsayed, D. W. Oard, and G. Namata. Resolving personal names in email using context expansion. *ACL*, pages 265–268. Association for Computational Linguistics, 2008. (Cited on page 25.)
 - [77] M. Eslami, A. Rickman, K. Vaccaro, A. Aleyasen, A. Vuong, K. Karahalios, K. Hamilton, and C. Sandvig. “i always assumed that i wasn’t really that close to [her]”: Reasoning about invisible algorithms in news feeds. *CHI*, pages 153–162. ACM, 2015. (Cited on page 2.)
 - [78] Y. Fang and M.-W. Chang. Entity linking on microblogs with spatial and temporal signals. *Transactions of the Association for Computational Linguistics*, 2:259–272, 2014. (Cited on page 20.)
 - [79] M. Färber, A. Rettinger, and B. El Asmar. On emerging entity detection. *EKAW*, pages 223–238. Springer,

8. Bibliography

2016. (Cited on page 21.)
- [80] I. P. Fellegi and A. B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64:1183–1210, 1969. (Cited on page 18.)
- [81] N. Fernandez, J. Fisteus, L. Sanchez, and E. Martin. Weblab: A co-occurrence-based approach to kbp 2010 entity-linking task. TAC. NIST, 2010. (Cited on page 19.)
- [82] S. Fertig, E. Freeman, and D. Gelernter. Lifestreams: An alternative to the desktop metaphor. CHI, pages 410–411. ACM, 1996. (Cited on page 26.)
- [83] S. Fertig, E. Freeman, and D. Gelernter. “finding and reminding” reconsidered. *SIGCHI Bull.*, 28(1): 66–69, 1996. (Cited on page 26.)
- [84] T. Finin, W. Murnane, A. Karandikar, N. Keller, J. Martineau, and M. Dredze. Annotating named entities in twitter data with crowdsourcing. CSLDAMT, pages 80–88. Association for Computational Linguistics, 2010. (Cited on page 145.)
- [85] J. R. Finkel, C. D. Manning, and A. Y. Ng. Solving the problem of cascading errors: Approximate bayesian inference for linguistic annotation pipelines. EMNLP, pages 618–626. Association for Computational Linguistics, 2006. (Cited on pages 19, 20, and 60.)
- [86] J. R. Firth. A Synopsis of Linguistic Theory, 1930-1955. *Studies in Linguistic Analysis*, pages 1–32, 1957. (Cited on page 65.)
- [87] J. R. Firth. *Papers in linguistics, 1934-1951*. Oxford University Press, 1957. (Cited on page 2.)
- [88] J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, F. Niu, C. Zhang, C. Re, and I. Soboroff. Building an entity-centric stream filtering test collection for TREC 2012. TREC. NIST, 2012. (Cited on page 36.)
- [89] J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, E. M. Voorhees, and I. Soboroff. Evaluating stream filtering for entity profile updates in TREC 2012, 2013, and 2014. TREC. NIST, 2014. (Cited on pages 35 and 36.)
- [90] J. H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002. (Cited on page 132.)
- [91] E. Gabrilovich and S. Markovitch. Wikipedia-based semantic interpretation for natural language processing. *Journal of Artificial Intelligence Research*, 34:443–498, 2009. (Cited on page 17.)
- [92] J. Gao, W. Yuan, X. Li, K. Deng, and J.-Y. Nie. Smoothing clickthrough data for web search ranking. SIGIR, pages 355–362. ACM, 2009. (Cited on page 23.)
- [93] C. Gărbacea, D. Odijk, D. Graus, I. Sijaranamual, and M. de Rijke. Combining multiple signals for semanticizing tweets: University of Amsterdam at #Microposts2015. #Microposts2015, pages 59–60, 2015. (Cited on page 12.)
- [94] F. Girardin, F. Calabrese, F. D. Fiore, C. Ratti, and J. Blat. Digital footprinting: Uncovering tourists with user-generated content. *IEEE Pervasive Computing*, 7:36–43, 2008. (Cited on page 1.)
- [95] Google. Building for the next moment. webpage, May 2015. (Cited on page 13.)
- [96] L. A. Granka. Measuring agenda setting with online search traffic: Influences of online and traditional media. In *2010 Annual Meeting of the American Political Science Association*, 2010. (Cited on page 58.)
- [97] D. Graus, T. Kenter, M. Bron, E. Meij, and M. De Rijke. Context-based entity linking-university of amsterdam at tac 2012. TAC, page 13. NIST, 2012. (Cited on pages 11 and 12.)
- [98] D. Graus, M.-H. Peetz, D. Odijk, O. de Rooij, and M. de Rijke. yourhistory–semantic linking for a personalized timeline of historic events. volume 4 of *LinkedUp Veni Competition on Linked and Open Data*. CEUR-WS, 2013. (Cited on page 12.)
- [99] D. Graus, Z. Ren, M. De Rijke, D. Van Dijk, H. Henseler, and N. Van Der Knaap. Semantic search in e-discovery: An interdisciplinary approach. volume 6 of *DESI V*, 2013. (Cited on page 12.)
- [100] D. Graus, D. Odijk, M. Tsagkias, W. Weerkamp, and M. de Rijke. Semanticizing search engine queries: The university of amsterdam at the ERD 2014 challenge. ERD, pages 69–74. ACM, 2014. (Cited on pages 11 and 12.)
- [101] D. Graus, M. Tsagkias, L. Buitinck, and M. de Rijke. Generating pseudo-ground truth for predicting new concepts in social streams. ECIR, pages 268–298. Springer, 2014. (Cited on page 11.)
- [102] D. Graus, D. van Dijk, M. Tsagkias, W. Weerkamp, and M. de Rijke. Recipient recommendation in enterprises using communication graphs and email content. SIGIR, pages 1079–1082. ACM, 2014.
- [103] D. Graus, P. N. Bennett, R. W. White, and E. Horvitz. Analyzing and predicting task reminders. UMAP, pages 7–15. ACM, 2016.
- [104] D. Graus, M. Tsagkias, W. Weerkamp, E. Meij, and M. de Rijke. Dynamic collective entity representations for entity ranking. WSDM, pages 595–604. ACM, 2016.
- [105] D. Graus, D. Odijk, and M. de Rijke. The birth of collective memories: Analyzing emerging entities in text streams. *arXiv:1701.04039*, 2017. (Cited on page 11.)
- [106] M. Grbovic, G. Halawi, Z. Karnin, and Y. Maarek. How many folders do you really need?: Classifying

-
- email into a handful of categories. CIKM, pages 869–878. ACM, 2014. (Cited on page 25.)
- [107] R. Grishman and B. Sundheim. Message understanding conference-6: A brief history. COLING, pages 466–471. Association for Computational Linguistics, 1996. (Cited on page 18.)
- [108] S. Guo, M.-W. Chang, and E. Kiciman. To link or not to link? a study on end-to-end tweet entity linking. NAACL, pages 1020–1030. Association for Computational Linguistics, 2013. (Cited on page 22.)
- [109] Y. Guo, W. Che, T. Liu, and S. Li. A graph-based method for entity linking. IJCNLP, pages 1010–1018, 2011. (Cited on page 19.)
- [110] I. Guy. Searching by talking: Analysis of voice queries on mobile web search. SIGIR, pages 35–44. ACM, 2016. (Cited on page 13.)
- [111] B. Hachey, W. Radford, J. Nothman, M. Honnibal, and J. R. Curran. Evaluating entity linking with wikipedia. *Artificial Intelligence*, 194:130–150, 2013. (Cited on page 21.)
- [112] M. Halbwachs. *La mémoire collective*. Albin Michel, 1950. (Cited on page 31.)
- [113] D. A. Hanauer, K. Wentzell, N. Laffel, and L. M. Laffel. Computerized Automated Reminder Diabetes System (CARDS): e-mail and SMS cell phone text messaging reminders to support diabetes management. *Diabetes Technology & Therapeutics*, 11(2):99–106, 2009. (Cited on page 26.)
- [114] F. Hasibi, K. Balog, and S. E. Bratsberg. Exploiting entity linking in queries for entity retrieval. ICTIR, pages 209–218. ACM, 2016. (Cited on pages 17 and 23.)
- [115] A. Hassan Awadallah, R. W. White, P. Pantel, S. T. Dumais, and Y.-M. Wang. Supporting complex search tasks. CIKM, pages 829–838. ACM, 2014. (Cited on page 14.)
- [116] J. He, M. de Rijke, M. Sevenster, R. van Ommering, and Y. Qian. Generating links to background knowledge: a case study using narrative radiology reports. CIKM, pages 1867–1876. ACM, 2011. (Cited on page 17.)
- [117] B. Hedin, S. Tomlinson, J. R. Baron, and D. W. Oard. Overview of the TREC 2009 legal track. TREC. NIST, 2010. (Cited on pages 25 and 104.)
- [118] H. Henseler. Network-based filtering for large email collections in e-discovery. *Artif. Intell. Law*, 2010. (Cited on page 104.)
- [119] J. L. Hicks, R. L. Marsh, and G. I. Cook. Task interference in time-based, event-based, and dual intention prospective memory conditions. *Journal of Memory and Language*, 53(3):430–444, 2005. (Cited on page 26.)
- [120] J. Hoffart, Y. Altun, and G. Weikum. Discovering emerging entities with ambiguous names. WWW, pages 385–396. ACM, 2014. (Cited on pages 21, 75, and 145.)
- [121] K. Hofmann, S. Whiteson, and M. de Rijke. A probabilistic method for inferring preferences from clicks. CIKM, pages 249–258, 2011. (Cited on page 79.)
- [122] K. Hofmann, A. Schuth, S. Whiteson, and M. de Rijke. Reusing historical interaction data for faster online learning to rank for ir. WSDM, pages 183–192. ACM, 2013. (Cited on pages 2 and 7.)
- [123] K. Hong, P. Pei, Y.-Y. Wang, and D. Hakkani-Tur. Entity ranking for descriptive queries. In *Spoken Language Technology Workshop*, pages 200–205. IEEE, 2014. (Cited on page 23.)
- [124] G.-J. Houben, G. McCalla, F. Pianesi, and M. Zancanaro, editors. *User Modeling, Adaptation, and Personalization: 17th International Conference, UMAP 2009, formerly UM and AH*, 2009. Springer Berlin Heidelberg. (Cited on page 7.)
- [125] M. Hu, A. Sun, and E.-P. Lim. Comments-oriented document summarization: understanding documents with readers’ feedback. SIGIR, pages 291–298. ACM, 2008. (Cited on page 67.)
- [126] M. Hu, S. Liu, F. Wei, Y. Wu, J. Stasko, and K.-L. Ma. Breaking news on twitter. CHI, pages 2751–2754. ACM, 2012. (Cited on page 59.)
- [127] Q. Hu, S. Bao, J. Xu, W. Zhou, M. Li, and H. Huang. Towards building effective email recipient recommendation service. SOLI, pages 398–403. IEEE, 2012. (Cited on page 25.)
- [128] R. S. Jeong. {FORZA} — digital forensics investigation framework that incorporate legal issues. *Digital Investigation*, 3, Supplement:29–36, 2006. (Cited on page 2.)
- [129] M. J. Intons-Peterson and J. Fournier. External and internal memory aids: When and how often do we use them? *Journal of Experimental Psychology: General*, 115(3):267–280, 1986. (Cited on page 26.)
- [130] H. Ji, R. Grishman, H. T. Dang, X. Li, K. Griffith, and J. Ellis. Overview of the TAC 2011 Knowledge Base Population Track. TAC. NIST, 2011. (Cited on pages 19 and 21.)
- [131] T. Joachims. Optimizing search engines using clickthrough data. KDD, pages 133–142. ACM, 2002. (Cited on page 24.)
- [132] H. Joho, L. A. Azzopardi, and W. Vanderbauwhede. A survey of patent users: An analysis of tasks, behavior, search functionality and system requirements. IliX, pages 13–24. ACM, 2010. (Cited on page 14.)
- [133] E. Kamar and E. Horvitz. Jogger: Models for context-sensitive reminding. AAMAS, pages 1089–1090.

8. Bibliography

- International Foundation for Autonomous Agents and Multiagent Systems, 2011. (Cited on pages 25 and 26.)
- [134] M. Kampf, E. Tessenow, D. Y. Kenett, and J. W. Kantelhardt. The detection of emerging trends using wikipedia traffic data and context networks. *PLoS ONE*, 10(12):1–19, 2016. (Cited on page 21.)
- [135] R. Kaptein, P. Serdyukov, A. de Vries, and J. Kamps. Entity ranking using wikipedia as a pivot. *CIKM*, pages 69–78. ACM, 2010. (Cited on page 23.)
- [136] P. N. Kapur, E. L. Glisky, and B. A. Wilson. Technological memory aids for people with memory deficits. *Neuropsychological Rehabilitation*, 14(1-2):41–60, 2004. (Cited on page 26.)
- [137] B. Keegan, D. Gergle, and N. Contractor. Hot off the wiki: Structures and dynamics of wikipedia’s coverage of breaking news events. *American Behavioral Scientist*, 57(5):595–622, 2013. (Cited on page 21.)
- [138] C. Kemp and K. Ramamohanarao. Long-term learning for web search engines. *PKDD*, pages 263–274. Springer, 2002. (Cited on pages 23, 99, and 145.)
- [139] T. Kenter, M. Wevers, P. Huijnen, and M. de Rijke. Ad hoc monitoring of vocabulary shifts over time. *CIKM*, pages 1191–1200. ACM, 2015. (Cited on page 145.)
- [140] M. A. Khalid, V. Jijkoun, and M. De Rijke. The impact of named entity normalization on information retrieval for question answering. *ECIR*, pages 705–710. Springer-Verlag, 2008. (Cited on pages 4 and 17.)
- [141] B. Klimt and Y. Yang. The enron corpus: A new dataset for email classification research. *ECML*, pages 217–226. Springer, 2004. (Cited on pages 24 and 107.)
- [142] Y. Koren, E. Liberty, Y. Maarek, and R. Sandler. Automatically tagging email by leveraging other users’ folders. *KDD*, pages 913–921. ACM, 2011. (Cited on page 24.)
- [143] Z. Kozareva. Bootstrapping named entity recognition with automatically generated gazetteer lists. *EACL-SRW*, pages 15–21. Association for Computational Linguistics, 2006. (Cited on page 22.)
- [144] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. *KDD*, pages 457–466. ACM, 2009. (Cited on page 21.)
- [145] R. Kumar and A. Tomkins. A characterization of online browsing behavior. *WWW*, pages 561–570. ACM, 2010. (Cited on page 77.)
- [146] M. Lamming and M. Flynn. “forget-me-not” intimate computing in support of human memory. *Cognitive Studies*, 2(1), 1995. (Cited on page 25.)
- [147] C.-J. Lee and W. B. Croft. Incorporating social anchors for ad hoc retrieval. *OAIR*, pages 181–188. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D’INFORMATIQUE DOCUMENTAIRE, 2013. (Cited on pages 23 and 79.)
- [148] J. Leskovec and E. Horvitz. Planetary-scale views on a large instant-messaging network. *WWW*, pages 915–924. ACM, 2008. (Cited on page 24.)
- [149] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. *KDD*, pages 497–506. ACM, 2009. (Cited on pages 58 and 59.)
- [150] A. Leuski. Email is a Stage: Discovering People Roles from Email Archives. *SIGIR*, pages 502–503. ACM, 2004. (Cited on page 25.)
- [151] S. C. Lewis and O. Westlund. Big data and journalism. *Digital Journalism*, 3:447–466, 2015. (Cited on page 2.)
- [152] F. Li, M. L. Lee, and W. Hsu. Entity profiling with varying source reliabilities. *KDD*, pages 1146–1155. ACM, 2014. (Cited on page 84.)
- [153] H. Li and J. Xu. Semantic matching in search. *Foundations and Trends in Information Retrieval*, 7: 343–469, 2014. (Cited on pages 14 and 17.)
- [154] T. W. Liao. Clustering of time series data—a survey. *Pattern Recognition*, 38:1857–1874, 2005. (Cited on page 39.)
- [155] T. Lin, Mausam, and O. Etzioni. No noun phrase left behind: Detecting and typing unlinkable entities. *EMNLP-CoNLL*, pages 893–903. Association for Computational Linguistics, 2012. (Cited on page 21.)
- [156] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning entity and relation embeddings for knowledge graph completion. *AAAI*, pages 2181–2187. AAAI Press, 2015. (Cited on page 24.)
- [157] J. Liu, P. Dolan, and E. R. Pedersen. Personalized news recommendation based on click behavior. *IUI*, pages 31–40. ACM, 2010. (Cited on page 7.)
- [158] J. Liu, E. Pedersen, and P. Dolan. Personalized news recommendation based on click behavior. *IUI*, pages 31–40. ACM, 2010. (Cited on page 7.)
- [159] T.-Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3 (3):225–331, 2009. (Cited on page 90.)
- [160] P. J. Ludford, D. Frankowski, K. Reily, K. Wilms, and L. Terveen. Because i carry my cell phone anyway: Functional location-based reminder applications. *CHI*, pages 889–898. ACM, 2006. (Cited on page 26.)

-
- [161] P. Lyman and H. R. Varian. How much information, 2003. [Online; accessed 16-August-2016]. (Cited on page 2.)
- [162] C. Macdonald, R. L. Santos, I. Ounis, and B. He. About learning models with multiple query-dependent features. *ACM TOIS*, 31(3):11:1–11:39, 2013. (Cited on pages 24, 79, and 85.)
- [163] T. W. Malone. How do people organize their desks?: Implications for the design of office information systems. *ACM TOIS*, 1(1):99–112, 1983. (Cited on page 26.)
- [164] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. (Cited on page 15.)
- [165] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*, chapter 8, pages 151–175. Cambridge University Press, 2008. (Cited on page 16.)
- [166] L. Manovich. Trending: the promises and the challenges of big social data. *Debates in the digital humanities*, pages 460–475, 2011. (Cited on page 2.)
- [167] G. Marchionini. Exploratory search: From finding to understanding. *Communications of the ACM*, 49: 41–46, 2006. (Cited on page 2.)
- [168] J. Markoff. Armies of expensive lawyers, replaced by cheaper software. *New York Times*, March 2011. (Cited on page 24.)
- [169] N. Marmasse and C. Schmandt. Location-aware information delivery with commotion. *Handheld and Ubiquitous Computing*, pages 157–171. Springer-Verlag, 2000. (Cited on page 26.)
- [170] K. Massoudi, M. Tsagkias, M. de Rijke, and W. Weerkamp. Incorporating query expansion and quality indicators in searching microblog posts. *ECIR*, pages 362–367. Springer, 2011. (Cited on page 67.)
- [171] A. McCallum, X. Wang, and A. Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *Journal of Artificial Intelligence Research*, 30:249–272, 2007. (Cited on page 107.)
- [172] M. McGee-Lennon, M. Wolters, R. McLachlan, S. Brewster, and C. Hall. Name that tune: musicons as reminders in the home. *CHI*, pages 2803–2806. ACM, 2011. (Cited on page 25.)
- [173] M. R. McGee-Lennon, M. K. Wolters, and S. Brewster. User-centred multimodal reminders for assistive living. *CHI*, pages 2105–2114. ACM, 2011. (Cited on page 26.)
- [174] P. McNamee and H. Dang. Overview of the TAC 2009 knowledge base population track. *TAC*. NIST, 2009. (Cited on page 18.)
- [175] O. Medelyan, I. Witten, and D. Milne. Topic indexing with wikipedia. In *Proceedings of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy*, pages 19–24. AAAI Press, 2008. (Cited on page 19.)
- [176] E. Meij, W. Weerkamp, and M. de Rijke. Adding semantics to microblog posts. *WSDM*, pages 563–572. ACM, 2012. (Cited on pages 5, 19, 20, 65, and 99.)
- [177] P. Mendes, M. Jakob, A. García-Silva, and C. Bizer. Dbpedia spotlight: Shedding light on the web of documents. *I-Semantics*, pages 1–8. ACM, 2011. (Cited on page 19.)
- [178] S. Meraz. Is there an elite hold? traditional media to social media agenda setting influence in blog networks. *Journal of Computer-Mediated Communication*, 14:682–707, 2009. (Cited on page 59.)
- [179] S. Meraz. Using time series analysis to measure intermedia agenda-setting influence in traditional media and political blog networks. *Journalism & Mass Communication Quarterly*, 88:176–194, 2011. (Cited on page 59.)
- [180] D. Metzler, J. Novak, H. Cui, and S. Reddy. Building enriched document representations using aggregated anchor text. *SIGIR*, pages 219–226. ACM, 2009. (Cited on pages 23 and 79.)
- [181] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. *CIKM*, pages 233–242. ACM, 2007. (Cited on pages 18, 19, and 140.)
- [182] D. Milne and I. Witten. Learning to link with Wikipedia. *CIKM*, pages 509–518, New York, 2008. ACM. (Cited on pages 19, 20, and 67.)
- [183] E. Minkov, R. C. Wang, and W. W. Cohen. Extracting personal names from emails: Applying named entity recognition to informal text. *HLT*, pages 443–450, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. (Cited on page 25.)
- [184] E. Minkov, W. W. Cohen, and A. Y. Ng. Contextual search and name disambiguation in email using graphs. *SIGIR*, pages 27–34. ACM, 2006. (Cited on page 25.)
- [185] G. Mishne and J. Lin. Twanchor text: A preliminary study of the value of tweets as anchor text. *SIGIR*, pages 1159–1160. ACM, 2012. (Cited on pages 79 and 83.)
- [186] S. Mizzaro. Relevance: The whole history. *Journal of the Association for Information Science and Technology*, 48:810–832, 1997. (Cited on page 1.)
- [187] A. Mohan, Z. Chen, and K. Q. Weinberger. Web-search ranking with initialized gradient boosted regression trees. *YLRC*, pages 77–89. JMLR.org, 2011. (Cited on page 90.)

8. Bibliography

- [188] M. Morita and Y. Shinoda. Information filtering based on user behavior analysis and best match text retrieval. SIGIR, pages 272–281. Springer-Verlag New York, Inc., 1994. (Cited on page 7.)
- [189] E. Morozov. The rise of data and the death of politics. *The Guardian*, 2014. (Cited on page 1.)
- [190] D. Mottin, T. Palpanas, and Y. Velegrakis. Entity ranking using click-log information. *Intelligent Data Analysis*, 17(5):837–856, 2013. (Cited on page 23.)
- [191] D. Müllner. fastcluster: Fast hierarchical, agglomerative clustering routines for r and python. *Journal of Statistical Software*, 53(1):1–18, 2013. (Cited on page 40.)
- [192] N. Nakashole, T. Tylenda, and G. Weikum. Fine-grained semantic typing of emerging entities. ACL, pages 1488–1497. The Association for Computer Linguistics, 2013. (Cited on page 21.)
- [193] F. Niu, C. Zhang, C. Ré, and J. Shavlik. Elementary: Large-scale knowledge-base construction via machine learning and statistical inference. *International Journal on Semantic Web and Information Systems*, 8:42–73, 2012. (Cited on page 20.)
- [194] M. Noll and C. Meinel. The metadata triumvirate: Social annotations, anchor texts and search queries. WI-IAT, pages 640–647. IEEE, 2008. (Cited on pages 23 and 83.)
- [195] J. Nothman, N. Ringland, W. Radford, T. Murphy, and J. R. Curran. Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence*, 194:151 – 175, 2013. (Cited on page 22.)
- [196] D. Oard, W. Webber, D. Kirsch, and S. Golitsynskiy. Avocado research email collection ldc2015t03. DVD, February 2015. (Cited on page 107.)
- [197] D. W. Oard and W. Webber. Information retrieval for e-discovery. *Foundations and Trends in Information Retrieval*, 7(2-3):99–237, 2013. (Cited on pages 2, 3, 4, 7, 99, 115, 139, and 145.)
- [198] B. O’Conaill and D. Frohlich. Timespace in the workplace: Dealing with interruptions. CHI, pages 262–263. ACM, 1995. (Cited on page 26.)
- [199] D. Odijk, E. Meij, and M. de Rijke. Feeding the second screen: Semantic linking based on subtitles. OAIR, pages 9–16. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D’INFORMATIQUE DOCUMENTAIRE, 2013. (Cited on pages 5, 19, 65, and 66.)
- [200] J. O’Neill, C. Privault, J.-M. Renders, V. Ciriza, and G. Bauduin. Disco: Intelligent help for document review. DESI III, 2009. (Cited on pages 7 and 103.)
- [201] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999. (Cited on page 104.)
- [202] C. Pal and A. McCallum. Cc prediction with graphical models. CEAS, 2006. (Cited on page 25.)
- [203] P. Pantel, E. Crestan, A. Borkovsky, A.-M. Popescu, and V. Vyas. Web-scale distributional similarity and entity set expansion. EMNLP, pages 938–947. Association for Computational Linguistics, 2009. (Cited on page 22.)
- [204] K. Partridge and B. Price. *Enhancing Mobile Recommender Systems with Activity Inference*, pages 307–318. UMAP. Springer Berlin Heidelberg, 2009. (Cited on page 25.)
- [205] C. Pentzold. Fixing the floating gap: The online encyclopaedia Wikipedia as a global memory place. *Memory Studies*, 2:255–272, 2009. (Cited on pages 4 and 31.)
- [206] J. R. Pérez-Agüera, J. Arroyo, J. Greenberg, J. P. Iglesias, and V. Fresno. Using bm25f for semantic search. SEMSEARCH, page Article 2. ACM, 2010. (Cited on page 24.)
- [207] S. Petrovic, M. Osborne, R. Mccreadie, C. Macdonald, and I. Ounis. Can twitter replace newswire for breaking news? ICWSM. AAAI Press, 2013. (Cited on pages 58 and 59.)
- [208] S. Pichai. Keynote. Google I/O, 05 2016. (Cited on page 13.)
- [209] D. Ploch, L. Hennig, E. De Luca, and S. Albayrak. Dai approaches to the tac-kbp 2011 entity linking task. TAC. NIST, 2011. (Cited on page 20.)
- [210] A. Qadir, M. Gamon, P. Pantel, and A. H. Awadallah. Activity modeling in email. NAACL-HLT, pages 1452–1462. Association for Computational Linguistics, 2016. (Cited on page 25.)
- [211] W. Radford, B. Hachey, M. Honnibal, J. Nothman, and J. Curran. Naive but effective NIL clustering baselines-CMCRC at TAC 2011. TAC. NIST, 2011. (Cited on page 19.)
- [212] S. Radicati. Email statistics report, 2016-2020 – executive summary, June 2016. (Cited on page 103.)
- [213] L. Ramshaw and M. Marcus. *Text Chunking using Transformation-Based Learning*, pages 157–176. Third Workshop on Very Large Corpora. Springer, 1999. (Cited on page 66.)
- [214] S. Rani and G. Sikka. Recent techniques of clustering of time series data: A survey. *International Journal of Computer Applications*, 52:1–9, 2012. (Cited on page 38.)
- [215] D. Rao, P. McNamee, and M. Dredze. Streaming cross document entity coreference resolution. COLING, pages 1050–1058. Association for Computational Linguistics, 2010. (Cited on page 145.)
- [216] D. Rao, P. McNamee, and M. Dredze. *Entity Linking: Finding Extracted Entities in a Knowledge Base*, pages 93–115. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. (Cited on pages 4 and 17.)
- [217] L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to

-
- wikipedia. HLT, pages 1375–1384. Association for Computational Linguistics, 2011. (Cited on pages 19 and 20.)
- [218] R. Reinanda, E. Meij, and M. de Rijke. Document filtering for long-tail entities. *CIKM*, pages 771–780. ACM, 2016. (Cited on page 21.)
- [219] B. J. Rhodes and T. Starnier. Remembrance agent: A continuously running automated information retrieval system. In *Proceedings of Practical Application of Intelligent Agents and Multi Agent Technology*, pages 487–495, 1996. (Cited on page 26.)
- [220] M. Richardson. Learning about the world through long-term query logs. *ACM Transactions on the Web*, 2(4):21:1–21:27, 2008. (Cited on page 24.)
- [221] A. Ritter, Mausam, O. Etzioni, and S. Clark. Open domain event extraction from twitter. *KDD*, pages 1104–1112. ACM, 2012. (Cited on page 20.)
- [222] S. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at trec–3. *TREC*, pages 109–126. NIST, 1995. (Cited on page 15.)
- [223] S. Robertson, H. Zaragoza, and M. Taylor. Simple bm25 extension to multiple weighted fields. *CIKM*, pages 42–49. ACM, 2004. (Cited on pages 24, 85, 89, 99, and 145.)
- [224] R. Rosenfeld. Two decades of statistical language modeling: Where do we go from here. *Proceedings of the IEEE*, 88(8):1270–1278, 2000. (Cited on page 15.)
- [225] M. Roth, A. Ben-David, D. Deutscher, G. Flysher, I. Horn, A. Leichtberg, N. Leiser, Y. Matias, and R. Merom. Suggesting friends using the implicit social graph. *KDD*, pages 233–242. ACM, 2010. (Cited on page 25.)
- [226] F. Scholer, H. E. Williams, and A. Turpin. Query association surrogates for web search: Research articles. *JASIST*, 55(7):637–650, 2004. (Cited on pages 23 and 81.)
- [227] A. Schuth. *Search Engines that Learn from Their Users*. PhD thesis, Informatics Institute, University of Amsterdam, 2016. (Cited on page 2.)
- [228] M. F. Schwartz and D. C. M. Wood. Discovering shared interests among people using graph analysis of global electronic mail traffic. *Communications of the ACM*, 36:78–89, 1993. (Cited on page 25.)
- [229] A. J. Sellen, G. Louie, J. E. Harris, and A. J. Wilkins. What brings intentions to mind? An in situ study of prospective memory. *Memory*, 5(4):483–507, 1997. (Cited on page 26.)
- [230] W. Shen, J. Wang, and J. Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *TKDE*, 27(2):443–460, 2014. (Cited on pages 18, 23, and 81.)
- [231] J. Shetty and J. Adibi. Discovering important nodes through graph entropy the case of enron email database. *LinkKDD*, pages 74–81. ACM, 2005. (Cited on page 25.)
- [232] A. Singhal and F. Pereira. Document expansion for speech retrieval. *SIGIR*, pages 34–41. ACM, 1999. (Cited on pages 23 and 79.)
- [233] I. M. Soboroff, I. Ounis, C. Macdonald, and J. Lin. Overview of the trec 2011 microblog track. *TREC*. NIST, 2012. (Cited on page 69.)
- [234] R. Socher, D. Chen, C. D. Manning, and A. Ng. Reasoning with neural tensor networks for knowledge base completion. *NIPS*, pages 926–934. Curran Associates, Inc., 2013. (Cited on page 24.)
- [235] W. M. Soon, H. T. Ng, and D. C. Y. Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27:521–544, 2001. (Cited on page 18.)
- [236] K. Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972. (Cited on page 15.)
- [237] J. Tasko, C. Görg, and Z. Liu. Jigsaw: Supporting investigative analysis through interactive visualization. *Information Visualization*, 7:118–132, 2008. (Cited on page 4.)
- [238] B. Suh, G. Convertino, E. H. Chi, and P. Pirolli. The singularity is not near: Slowing growth of wikipedia. *WikiSym*, pages 8:1–8:10. ACM, 2009. (Cited on page 31.)
- [239] K. M. Svore and C. J. Burges. A machine learning approach for improved bm25 retrieval. *CIKM*, pages 1811–1814. ACM, 2009. (Cited on page 24.)
- [240] G. Szabo and B. A. Huberman. Predicting the popularity of online content. *Communications of the ACM*, 53(8):80–88, 2010. (Cited on page 89.)
- [241] J. Tague, M. Nelson, and H. Wu. Problems in the simulation of bibliographic retrieval systems. *SIGIR*, pages 236–255. Butterworth & Co., 1980. (Cited on page 22.)
- [242] O. Tene and J. Polonetsky. Big data for all: Privacy and user control in the age of analytics. *Social Science Research Network Working Paper Series*, 11:239–273, 2013. (Cited on page 1.)
- [243] E. F. Tjong Kim Sang. Introduction to the conll-2002 shared task: Language-independent named entity recognition. *COLING*, pages 1–4. Association for Computational Linguistics, 2002. (Cited on page 18.)
- [244] M. Tsagkias, W. Weerkamp, and M. de Rijke. News comments: Exploring, modeling, and online prediction. *ECIR*, pages 191–203. Springer, 2010. (Cited on page 89.)

8. Bibliography

- [245] Y. Tu, L. Chen, M. Lv, Y. Ye, W. Huang, and G. Chen. ireminder: An intuitive location-based reminder that knows where you are going. *International Journal of Human–Computer Interaction*, 29(12):838–850, 2013. (Cited on page 26.)
- [246] G. Tuchman. Objectivity as strategic ritual: An examination of newsmen’s notions of objectivity. *American Journal of Sociology*, 77:660–679, 1972. (Cited on page 58.)
- [247] Z. Tufekci. Algorithmic harms beyond facebook and google: Emergent challenges of computational agency symposium essays. *Colorado Technology Law Journal*, 13:203, 2015. (Cited on page 1.)
- [248] T. Tyenda, R. Angelova, and S. Bedathur. Towards time-aware link prediction in evolving social networks. SNA-KDD, page Article 9. ACM, 2009. (Cited on page 146.)
- [249] P. Vakkari. A theory of the task-based information retrieval process: a summary and generalisation of a longitudinal study. *Journal of Documentation*, 57:44–60, 2001. (Cited on page 14.)
- [250] D. van Dijk, H. Henseler, and M. de Rijke. Semantic search in e-discovery. DESI IV, 2011. (Cited on pages 4 and 139.)
- [251] D. van Dijk, D. Graus, Z. Ren, H. Henseler, and M. de Rijke. Who is involved? semantic search for e-discovery. DESI VI, 2015. (Cited on page 12.)
- [252] A.-M. Vercoustre, J. A. Thom, and J. Pehcevski. Entity ranking in wikipedia. SAC, pages 1101–1106. ACM, 2008. (Cited on page 23.)
- [253] M. Vlachos, C. Meek, Z. Vagena, and D. Gunopulos. Identifying similarities, periodicities and bursts for online search queries. SIGMOD, pages 131–142. ACM, 2004. (Cited on pages 38, 39, and 43.)
- [254] U. Von Luxburg, R. C. Williamson, and I. Guyon. Clustering: Science or art? UTLW, pages 65–79. JMLR.org, 2011. (Cited on pages 40, 60, and 140.)
- [255] N. Voskarides, D. Odijk, M. Tsagkias, W. Weerkamp, and M. de Rijke. Query-dependent contextualization of streaming data. ECIR, pages 706–712. Springer, 2014. (Cited on page 21.)
- [256] D. P. Wallace and C. Van Fleet. From the editors: The democratization of information? Wikipedia as a reference resource. *Reference & User Services Quarterly*, 45:100–103, 2005. (Cited on page 4.)
- [257] J. H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963. (Cited on page 40.)
- [258] W. Weerkamp and M. de Rijke. Credibility-inspired ranking for blog post retrieval. *Information Retrieval*, 15(3-4):243–277, 2012. (Cited on page 67.)
- [259] T. Westerveld, W. Kraaij, and D. Hiemstra. Retrieving web pages using content, links, urls and anchors. TREC. NIST, 2001. (Cited on pages 23, 79, and 82.)
- [260] R. W. White and S. M. Drucker. Investigating behavioral variability in web search. WWW, pages 21–30. ACM, 2007. (Cited on page 24.)
- [261] R. W. White, N. P. Tatonetti, N. H. Shah, R. B. Altman, and E. Horvitz. Web-scale pharmacovigilance: listening to signals from the crowd. *Journal of the American Medical Informatics Association*, 20(3): 404–408, 2013. (Cited on page 2.)
- [262] R. W. White, M. Richardson, and W.-t. Yih. Questions vs. queries in informational search tasks. WWW, pages 135–136. ACM, 2015. (Cited on page 14.)
- [263] Wikipedia. Hillary clinton email controversy — wikipedia, the free encyclopedia, 2016. [Online; accessed 19-September-2016]. (Cited on page 2.)
- [264] Wikipedia. Enron scandal — wikipedia, the free encyclopedia, 2016. [Online; accessed 19-September-2016]. (Cited on page 2.)
- [265] Wikipedia. Panama papers — wikipedia, the free encyclopedia, 2016. [Online; accessed 19-September-2016]. (Cited on page 2.)
- [266] Wikipedia. Information published by wikileaks — wikipedia, the free encyclopedia, 2016. [Online; accessed 19-September-2016]. (Cited on page 2.)
- [267] Wikipedia. Wikipedia:size of wikipedia — wikipedia, the free encyclopedia, 2016. [Online; accessed 12-September-2016]. (Cited on page 4.)
- [268] J. wook Ahn, P. Brusilovsky, J. Grady, D. He, and R. Florian. Semantic annotation based exploratory search for information analysts. *Information Processing & Management*, 46(4):383 – 402, 2010. (Cited on pages 3, 4, 17, 77, and 139.)
- [269] D. Wu, W. S. Lee, N. Ye, and H. L. Chieu. Domain adaptive bootstrapping for named entity recognition. EMNLP, pages 1523–1532. Association for Computational Linguistics, 2009. (Cited on page 22.)
- [270] M. Wu, D. Hawking, A. Turpin, and F. Scholer. Using anchor text for homepage and topic distillation search tasks. *JASIST*, 63(6):1235–1255, 2012. (Cited on pages 23 and 79.)
- [271] G.-R. Xue, H.-J. Zeng, Z. Chen, Y. Yu, W.-Y. Ma, W. Xi, and W. Fan. Optimizing web search using web click-through data. CIKM, pages 118–126. ACM, 2004. (Cited on page 23.)
- [272] T. Yasseri, R. Sumi, and J. Kertész. Circadian patterns of wikipedia editorial activity: A demographic

-
- analysis. *PLoS ONE*, 7(1):1–8, 2012. (Cited on page 21.)
- [273] T. Yasseri, R. Sumi, A. Rung, A. Kornai, and J. Kertész. Dynamics of conflicts in wikipedia. *PLoS ONE*, 7(6):1–12, 2012. (Cited on pages 21 and 64.)
- [274] K. Yelupula and S. Ramaswamy. Social network analysis for email classification. *ACM-SE 46*, pages 469–474. ACM, 2008. (Cited on page 25.)
- [275] E. Yom-Tov and E. Gabrilovich. Postmarket drug surveillance without trial costs: Discovery of adverse drug reactions through large-scale analysis of web search queries. *Journal of Medical Internet Research*, 15:e124, 2013. (Cited on page 2.)
- [276] W. Youyou, M. Kosinski, and D. Stillwell. Computer-based personality judgments are more accurate than those made by humans. *PNAS*, 112:1036–1040, 2015. (Cited on page 1.)
- [277] H. Zaragoza, N. Craswell, M. Taylor, S. Saria, and S. Robertson. Microsoft Cambridge at TREC-13: Web and hard tracks. TREC. NIST, 2004. (Cited on page 24.)
- [278] W. Zhang, J. Su, C. L. Tan, and W. T. Wang. Entity linking leveraging: Automatically generated annotation. *COLING*, pages 1290–1298. Association for Computational Linguistics, 2010. (Cited on page 22.)
- [279] Y. Zhou, L. Nie, O. Rouhani-Kalleh, F. Vasile, and S. Gaffney. Resolving surface forms to wikipedia topics. *COLING*, pages 1335–1343. Association for Computational Linguistics, 2010. (Cited on page 22.)

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 Business Conversations*
- 4 Dennis Breuker (UM) *Memory versus Search in Games*
- 5 E. W. Oskamp (RUL) *Computerondersteuning bij Straftoemeting*

1999

- 1 Mark Sloof (VUA) *Physiology of Quality Change Modelling: Automated modelling of*
- 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*
- 6 Niek J. E. Wijngaards (VUA) *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*

2000

- 1 Frank Niessink (VUA) *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*
- 4 Geert de Haan (VUA) *ETAG, A Formal Model of Competence Knowledge for User Interface*
- 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 Coupé (EUR) *Sensitivity Analysis 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 (VUA) *Processing Structured Hypermedia: A Matter of Style*
- 6 Martijn van Welie (VUA) *Task-based User Interface Design*
- 7 Bastiaan Schonhage (VUA) *Divas: Architectural Perspectives on Information Visualization*
- 8 Pascal van Eck (VUA) *A Compositional Semantic Structure for Multi-Agent Systems Dynamics*
- 9 Pieter Jan 't Hoen (RUL) *Towards Distributed Development of Large Object-Oriented Models*
- 10 Maarten Sierhuis (UvA) *Modeling and Simulating Work Practice*
- 11 Tom M. van Engers (VUA) *Knowledge Management*

2002

- 1 Nico Lassing (VUA) *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 (VUA) *The Private Cyberspace Modeling Electronic*
- 6 Laurens Mommers (UL) *Applied legal epistemology: Building a knowledge-based ontology of*
- 7 Peter Boncz (CWI) *Monet: A Next-Generation DBMS Kernel For Query-Intensive*
- 8 Jaap Gordijn (VUA) *Value Based Requirements Engineering: Exploring Innovative*
- 9 Willem-Jan van den Heuvel (KUB) *Integrating Modern Business Applications with Objectified Legacy*
- 10 Brian Sheppard (UM) *Towards Perfect Play of Scrabble*
- 11 Wouter C. A. Wijngaards (VUA) *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 (VUA) *The Anatomy of Design: Foundations, Models and Applications*
- 17 Stefan Manegold (UvA) *Understanding, Modeling, and Improving Main-Memory Database Performance*

2003

- 1 Heiner Stuckenschmidt (VUA) *Ontology-Based Information Sharing in Weakly Structured Environments*
- 2 Jan Broersen (VUA) *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 the interaction 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 Weerd (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 (VUA) *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 politieke gegevensuitwisseling en digitale expertise*
- 9 Martin Caminada (VUA) *For the Sake of the Argument: explorations into argument-based reasoning*
- 10 Suzanne Kabel (UvA) *Knowledge-rich indexing of learning-objects*
- 11 Michel Klein (VUA) *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 (VUA) *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 (VUA) *Storage, Querying and Inferencing for Semantic Web Languages*
 - 10 Anders Bouwer (UvA) *Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments*
 - 11 Elth Ogston (VUA) *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 (VUA) *Web-Service configuration on the Semantic Web: Exploring how semantics meets pragmatics*
 - 15 Tibor Bosse (VUA) *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*
 - 22 Eelco Herder (UT) *Forward, Back and Home Again: Analyzing User Behavior on the Web*
 - 23 Mohamed Wahdan (UM) *Automatic Formulation of the Auditor's Opinion*
 - 24 Ronny Siebes (VUA) *Semantic Routing in Peer-to-Peer Systems*
 - 25 Joeri van Ruth (UT) *Flattening Queries over Nested Data Types*
 - 26 Bert Bongers (VUA) *Interactivation: Towards an e-cology of people, our technological environment, and the arts*
 - 27 Henk-Jan Lebbink (UU) *Dialogue and Decision Games for Information Exchanging Agents*
 - 28 Johan Hoorn (VUA) *Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change*
 - 29 Rainer Malik (UU) *CONAN: Text Mining in the Biomedical Domain*
 - 30 Carsten Riggelsen (UU) *Approximation Methods for Efficient Learning of Bayesian Networks*
 - 31 Stacey Nagata (UU) *User Assistance for Multitasking with Interruptions on a Mobile Device*
 - 32 Valentin Zhizhkhun (UvA) *Graph transformation for Natural Language Processing*
 - 33 Birna van Riemsdijk (UU) *Cognitive Agent Programming: A Semantic Approach*
 - 34 Marina Velikova (UvT) *Monotone models for prediction in data mining*
 - 35 Bas van Gils (RUN) *Aptness on the Web*
 - 36 Paul de Vrieze (RUN) *Fundamentals of Adaptive Personalisation*
 - 37 Ion Juvina (UU) *Development of Cognitive Model for Navigating on the Web*
 - 38 Laura Hollink (VUA) *Semantic Annotation for Retrieval of Visual Resources*
 - 39 Madalina Drugan (UU) *Conditional log-likelihood MDL and Evolutionary MCMC*
 - 40 Vojkan Mihajlovic (UT) *Score Region Algebra: A Flexible Framework for Structured Information Retrieval*
 - 41 Stefano Bocconi (CWI) *Vox Populi: generating video documentaries from semantically annotated media repositories*
 - 42 Borkur Sigurbjornsson (UvA) *Focused Information Access using XML Element Retrieval*
- 2006**
- 1 Samuil Angelov (TUE) *Foundations of B2B Electronic Contracting*
 - 2 Cristina Chisalita (VUA) *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 (VUA) *Building Web Service Ontologies*
 - 5 Cees Pierik (UU) *Validation Techniques for Object-Oriented Proof Outlines*
 - 6 Ziv Baida (VUA) *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*
- 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 (VUA) *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 (VUA) *Modeling of Change in Multi-Agent Organizations*
- 9 David Mobach (VUA) *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 (VUA) *Empirical Investigations in Software Process Improvement*
- 2 Alexei Sharpanskykh (VUA) *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 (VUA) *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 Bunnigen (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 (VUA) *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 (VUA) *Using background knowledge in ontology matching*
- 25 Geert Jonker (UU) *Efficient and Equitable Exchange in Air Traffic Management Plan Repair using Spender-signed Currency*

2008

- 1 Katalin Boer-Sorbán (EUR) *Agent-Based Simulation of Financial Markets: A modular, continuous-time approach*

- 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 (VUA) *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 (VUA) *Evaluating Ontology-Alignment Techniques*
 - 3 Hans Stol (UvT) *A Framework for Evidence-based Policy Making Using IT*
 - 4 Josephine Nabukanya (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 (VUA) *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) *Operating Guidelines for Services*
 - 13 Steven de Jong (UM) *Fairness in Multi-Agent Systems*
 - 14 Maksym Korotkiy (VUA) *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: Controlling 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 (VUA) *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 (VUA) *Architectural Knowledge Management: Supporting Architects and Auditors*
 - 33 Khiat 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 Drachslar (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 Bereznyy (UvT) *Digital Analysis of Paintings*
 - 42 Toine Bogers (UvT) *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 Children*
 - 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 (VUA) *Automated Web Service Reconfiguration*
 - 15 Lianne Bodenstaff (UT) *Managing Dependency Relations in Inter-Organizational Models*
 - 16 Sizzo Verwer (TUD) *Efficient Identification of Timed Automata, theory and practice*
 - 17 Spyros Kotoulas (VUA) *Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications*
 - 18 Charlotte Gerritsen (VUA) *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 Zulfiqar Ali Memon (VUA) *Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective*
 - 25 Ying Zhang (CWI) *XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines*
 - 26 Marten Voulon (UL) *Automatisch contracteren*
 - 27 Arne Koopman (UU) *Characteristic Relational Patterns*
 - 28 Stratos Idreos (CWI) *Database Cracking: Towards Auto-tuning Database Kernels*
 - 29 Marieke van Erp (UvT) *Accessing Natural History: Discoveries in data cleaning, structuring, and retrieval*
 - 30 Victor de Boer (UvA) *Ontology Enrichment from Heterogeneous Sources on the Web*
 - 31 Marcel Hiel (UvT) *An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems*
 - 32 Robin Aly (UT) *Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval*
 - 33 Teduh Dirgahayu (UT) *Interaction Design in Service Compositions*
 - 34 Dolf Trieschnigg (UT) *Proof of Concept: Concept-based Biomedical Information Retrieval*
 - 35 Jose Janssen (OU) *Paving the Way for Lifelong Learning: Facilitating competence development through a learning path specification*
 - 36 Niels Lohmann (TUE) *Correctness of services and their composition*
 - 37 Dirk Fahland (TUE) *From Scenarios to components*
 - 38 Ghazanfar Farooq Siddiqui (VUA) *Integrative modeling of emotions in virtual agents*
 - 39 Mark van Assem (VUA) *Converting and Integrating Vocabularies for the Semantic Web*
 - 40 Guillaume Chaslot (UM) *Monte-Carlo Tree Search*
 - 41 Sybren de Kinderen (VUA) *Needs-driven service bundling in a multi-supplier setting: the computational e3-service approach*
 - 42 Peter van Kranenburg (UU) *A Computational Approach to Content-Based Retrieval of Folk Song Melodies*

- 43 Pieter Bellekens (TUE) *An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain*
 - 44 Vasilios Andrikopoulos (UvT) *A theory and model for the evolution of software services*
 - 45 Vincent Pijpers (VUA) *e3alignment: Exploring Inter-Organizational Business-ICT Alignment*
 - 46 Chen Li (UT) *Mining Process Model Variants: Challenges, Techniques, Examples*
 - 47 Jahn-Takeshi Saito (UM) *Solving difficult game positions*
 - 48 Bouke Huurnink (UvA) *Search in Audiovisual Broadcast Archives*
 - 49 Alia Khairia Amin (CWI) *Understanding and supporting information seeking tasks in multiple sources*
 - 50 Peter-Paul van Maanen (VUA) *Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention*
 - 51 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*
 - 5 Base van der Raadt (VUA) *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. Multi-agent 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 (VUA) *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 (VUA) *Analysis and Validation of Models for Trust Dynamics*
 - 26 Matthijs Aart Pontier (VUA) *Virtual Agents for Human Communication: Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots*
 - 27 Aniel Bhulai (VUA) *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 (VUA) *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 (VUA) *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 (VUA) *Adaptivity, emotion, and Rationality in Human and Ambient Agent Models*
- 3 Adam Vanya (VUA) *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 (VUA) *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 (VUA) *Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes*
- 16 Fiemke Both (VUA) *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 (VUA) *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 (VUA) *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 (VUA) *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 (VUA) *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) *Ontology driven Enterprise Information Systems Engineering*
- 50 Jeroen de Jong (TUD) *Heuristics in Dynamic Scheduling: a practical framework with a case study in elevator dispatching*
- 11 Evangelos Pournaras (TUD) *Multi-level Reconfigurable Self-organization in Overlay Services*
- 12 Marian Razavian (VUA) *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*
- 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 (VUA) *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 Börner (OUN) *Ambient Learning Displays*

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 (VUA) *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 (VUA) *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*

- 38 Eelco den Heijer (VUA) *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 Tuliyo (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 Reijssen (UU) *Knowledge Perspectives on Advancing Dynamic Capability*
 - 6 Damian Tamburri (VUA) *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 (VUA) *Service Value Networks*
 - 11 Janneke van der Zwaan (TUD) *An Empathic Virtual Buddy for Social Support*
 - 12 Willem van Willigen (VUA) *Look Ma, No Hands: Aspects of Autonomous Vehicle Control*
 - 13 Arlette van Wissen (VUA) *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 (VUA) *Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare*
 - 16 Krystyna Milian (VUA) *Supporting trial recruitment and design by automatically interpreting eligibility criteria*
 - 17 Kathrin Dentler (VUA) *Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability*
 - 18 Mattijs Ghijzen (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 Cassidy 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 (VUA) *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 (VUA) *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 (VUA) *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 (CWITUD) *Contextual Multidimensional Relevance Models*
 - 43 Kevin Vlaanderen (UU) *Supporting Process Improvement using Method Increments*
 - 44 Paulien Meesters (UvT) *Intelligent Blauw: 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 (VUA) *Modi Operandi of Social Network Dynamics: The Effect of Context on Scientific Collaboration Networks*
 - 13 Giuseppe Procaccianti (VUA) *Energy-Efficient Software*
 - 14 Bart van Straalen (UT) *A cognitive approach to modeling bad news conversations*
 - 15 Klaas Andries de Graaf (VUA) *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 Zheming Zhu (UT) *Co-occurrence Rate Networks*
 - 23 Luit Gazendam (VUA) *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 column 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 (VUA) *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 (VUA) *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 (VUA) *Trusting Crowd-sourced 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 (VUA) *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 (VUA) *Towards Embodied Evolution of Robot Organisms*
- 18 Albert Meroño Peñuela (VUA) *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 Céleri (UT) *From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground*
- 22 Grace Lewis (VUA) *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 (VUA) *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 Cyber-crime*
- 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 Vandić (EUR) *Intelligent Information Systems for Web Product Search*
- 7 Roel Bertens (UU) *Insight in Information: from Abstract to Anomaly*
- 8 Rob Konijn (VUA) *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 (UT) *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, Radboud University (RUN) *Memory-Based Text Correction*
- 16 Aleksandr Chuklin (UvA) *Understanding and Modeling Users of Modern Search Engines*
- 17 Daniel Dimov (VUA) *Crowdsourced Online Dispute Resolution*
- 18 Ridho Reinanda (UvA) *Entity Associations for Search*
- 19 Jeroen Vuurens (TUD) *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 (VUA) *Logics for causal inference under uncertainty*
- 23 David Graus (UvA) *Entities of Interest — Discovery in Digital Traces*

Tegenwoordig laten we continu — en soms zonder het te weten — digitale sporen achter, door online content te browsen, *sharen*, *liken*, doorzoeken, bekijken, of beluisteren. Geaggregeerd kunnen deze digitale sporen waardevolle inzichten bieden in ons gedrag, onze voorkeuren, bezigheden, en eigenschappen. Terwijl veel mensen zich zorgen maken om de hiermee gepaarde bedreiging van onze privacy, heeft het op grote schaal verzamelen en gebruiken van onze digitale sporen ook veel goeds gebracht. Bijvoorbeeld de toegang tot ongekende hoeveelheden informatie en kennis die de zoekmachines — al lerende van hun gebruikers — ons verschaffen, of het ontdekken van nieuwe bijwerkingen van medicijnen door de zoekopdrachten die bij een zoekmachine binnenkomen te analyseren.

Of het nu in online diensten, journalistiek, digitaal sporenonderzoek, of wetenschap is, men richt zich steeds meer op digitale sporen om nieuwe informatie te ontdekken. Neem bijvoorbeeld het Enron schandaal, de controverse rond het emailgebruik van Hillary Clinton, of de Panama Papers: voorbeelden van zaken die draaien rond het analyseren, doorzoeken, onderzoeken, en binnenstebuiten keren van grote hoeveelheden digitale sporen om tot nieuwe inzichten, kennis, en informatie te komen.

Dit proefschrift gaat over het ontdekkingsproces in grootschalige verzamelingen digitale sporen. Het proefschrift bevindt zich op de kruising van zoekmachinetechnologie, taaltechnologie, en toegepaste *machine learning*, en presenteert computationele methoden die ondersteuning bieden aan het doorzoeken en inzichtelijk maken van grootschalige verzamelingen digitale sporen. We richten ons op tekstuele digitale sporen, zoals emails, en social media. Daarnaast richten we ons op twee aspecten die centraal staan in het ontdekkingsproces.

Allereerst richten we ons op de inhoud van digitale sporen. Onze onderzoekssubjecten zijn de personen, plaatsen, en organisaties die worden genoemd in tekstuele digitale sporen. Deze zogenaamde *real-world entities* zijn van centraal belang in het doorzoeken en inzichtelijk maken van de inhoud van digitale sporen. In dit deel richten we ons op het analyseren, herkennen, en beter vindbaar maken van nieuwe entiteiten die opkomen in digitale sporen.

In het tweede deel richten we ons op de producenten van digitale sporen. Niet de inhoud, maar de context waarin de digitale sporen tot stand komen staat hierbij centraal. Ons doel is toekomstige activiteiten te voorspellen door gebruik te maken van digitale sporen. We tonen aan dat we de ontvanger van een email kunnen voorspellen door gebruik te maken van de communicatie netwerken van emailverkeer. Daarnaast tonen we aan dat we kunnen voorspellen wanneer iemand van plan is een bepaalde taak uit te voeren, door gebruik te maken van geaggregeerde interactie-data met een persoonlijke assistent-applicatie.

In the era of big data, we continuously — and at times unknowingly — leave behind digital traces, by browsing, sharing, posting, liking, searching, watching, and listening to online content. When aggregated, these digital traces can provide powerful insights into the behavior, preferences, activities, and traits of people. While many have raised privacy concerns around the use of aggregated digital traces, it has undisputedly brought us many advances, from the search engines that learn from their users and enable our access to unforeseen amounts of data, knowledge, and information, to, e.g., the discovery of previously unknown adverse drug reactions from search engine logs.

Whether in online services, journalism, digital forensics, law, or research, we increasingly set out to exploring large amounts of digital traces to discover new information. Consider for instance, the Enron scandal, Hillary Clinton’s email controversy, or the Panama papers: cases that revolve around analyzing, searching, investigating, exploring, and turning upside down large amounts of digital traces to gain new insights, knowledge, and information. This *discovery* task is at its core about “finding evidence of activity in the real world.”

This dissertation revolves around *discovery in digital traces*, and sits at the intersection of Information Retrieval, Natural Language Processing, and applied Machine Learning. We propose computational methods that aim to support the exploration and sense-making process of large collections of digital traces. We focus on textual traces, e.g., emails and social media streams, and address two aspects that are central to discovery in digital traces.

In the first part, we focus on the textual content of digital traces. Here, our entities of interest are the people, places, and organizations that appear in digital traces. These so-called real-world entities are central to enabling exploratory search in digital traces. More specifically, we analyze, predict and show how to improve retrieval of newly *emerging* entities, i.e., previously unknown entities that surface in digital traces.

In the second part, our entities of interest are the people who leave behind the digital traces. Here, we focus on the contexts in which digital traces are created, and aim to predict users’ future activity, by leveraging their historic digital traces. We show we can predict email recipients by leveraging the email communication network, and we show we can predict when users plan to execute activities, by leveraging logs of interactions with an intelligent assistant.

ENTITIES OF INTEREST DISCOVERY IN DIGITAL TRACES

In the era of big data, we continuously – and at times unknowingly – leave behind *digital traces*, by browsing, sharing, posting, liking, searching, watching, and listening to online content. Aggregated, these digital traces can provide powerful insights into the behavior, preferences, activities, and traits of people. While many have raised privacy concerns around the use of aggregated digital traces, it has undisputedly brought us many advances, from the search engines that enable our access to unforeseen amounts of data, knowledge, and information, to, e.g., the discovery of previously unknown adverse drug reactions from search engine logs.

Whether in online services, journalism, digital forensics, law, or research, we increasingly set out to exploring large amounts of digital traces to discover new information. Consider for instance, the Enron scandal, Hillary Clinton's email controversy, or the Panama Papers: cases that revolve around analyzing, searching, investigating, exploring, and turning upside down large amounts of digital traces to gain new insights, knowledge, and information. This *discovery* task is at its core about "finding evidence of activity in the real world."

This dissertation revolves around *discovery in digital traces*. We propose computational methods that aim to support the exploration and sense-making process of large collections of textual digital traces, e.g., emails and social media streams. We address methods for analyzing the textual *content* of digital traces, and the *contexts* in which they are created, with the goal of predicting people's future activity, by leveraging their historic digital traces.



ISBN 978-94-6182-800-2