UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Entity associations for search

Reinanda, R.

Publication date
2017

Document Version
Final published version

License
Other

Link to publication

Citation for published version (APA):
Reinanda, R. (2017). *Entity associations for search*. [Thesis, externally prepared, Universiteit van Amsterdam].

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:27 Mar 2025

Entities are commonly defined as things that have existence in the real-world, such as persons, organizations, and locations. Being identifiable, independent objects, they make nice shortcuts to access information. An entity rarely stands on its own, however. Associations with other objects such as related entities, attributes, and topics are also of importance when working with an entity. In this thesis, we investigate the broad problem of computing entity associations for search. Specifically, we consider three types of entity association: **entity-entity**, **entity-document**, and **entity-aspect** associations. We touch upon various domains, starting with specific domains such as the humanities and business, and ending in Web search. This thesis contributes new task formalizations, algorithms, and insights on computing entity associations in the context of search. Insights gained from this thesis will help address entity-oriented information access challenges in various domains.

Entity Associations for Search

Ridho Reinanda

Entity Associations for Search

Ridho Reinanda

Entity Associations for Search

Ridho Reinanda

Entity Associations for Search

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 donderdag 11 mei 2017, te 10:00 uur

door

Ridho Reinanda

geboren te Padang, Indonesië

Promotiecommissie

Promotor:		
	Prof. dr. M. de Rijke	Universiteit van Amsterdam
Co-promotor:		
	Dr. G.I.J. Steijlen	Koninklijk Instituut voor Taal, Land en Volkenkunde
Overige leden:		
	Prof. dr. F.M.G. de Jong	Universiteit Twente
	Dr. E. Kanoulas	Universiteit van Amsterdam
	Prof. dr. G. van Klinken	Universiteit van Amsterdam
	Dr. A. Purwarianti	Institut Teknologi Bandung
	Prof. dr. M. Worring	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

SIKS Dissertation Series No. 2017-18
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Copyright © 2017 Ridho Reinanda, Amsterdam, The Netherlands

Cover by Mostafa Dehghani
Printed by Off Page, Amsterdam
Summary translated by Tom Kenter

ISBN: 978-94-6182-788-3

Acknowledgements

I started my PhD journey nearly four and a half years ago. It has been a wonderful experience that has allowed me to meet many people over the years and learn from them. I would like to express my sincerest gratitude to everyone who has helped me in this journey.

First and foremost, I want to thank my supervisor, Maarten. You taught me how to be a researcher: finding interesting problems, formulating those problems, coming up with solutions, and writing down the results. Your research and non-research related advice has been instrumental over the past few years. More often than not, they led to all sorts of interesting results and opportunities. Thank you, Maarten.

Fridus, thanks for sharing your research experience and for looking out for me. Your guidance, sense of humor, and tendency to always look for the positives are very important in ensuring that I made it through to the end of my PhD journey.

I am honored to have Franciska de Jong, Evangelos Kanoulas, Gerry van Klinken, Ayu Purwarianti, and Marcel Worring serving in my doctoral committee.

Edgar, thanks for picking up your daily supervisor role every once in a while (long after you had left the university). I learned a lot from your vast research experience, curiosity, and enthusiasm. I always come out motivated after our ad-hoc meetings. Looking forward to our collaborations in the near future.

I want to thank the members of the Elite Network Shifts project: Andrea, Ayu, Franciska, Fridus, Gerry, Jacky, Vincent, and Wolfgang. The project gave rise to many discussions and it has helped me to identify my own research interests. Thanks, my fellow core researchers Jacky and Vincent, for your critical comments and interesting questions. I have learned a lot from our many discussions and collaborations. Thank you, Gerry, for showing me how research is conducted in your field. Your curiosity and optimism have always motivated me to work on the project. Bu Ayu, thanks for introducing me to this project.

ILPS is a wonderful environment to do research. I want to thank the ILPS members who have shared their time with me: Abdo, Aleksandr, Aleksey, Anne, Artem, Caroline, Christophe, Chuan, Cristina, Daan, Damien, David, David, Evgeny, Fei, Fons, Hamid, Harrie, Hendrike, Hosein, Ilya, Isaac, Ivan, Julia, Katja, Katya, Ke, Lars, Manos, Marc, Marlies, Marzieh, Masrour, Mostafa, Nikos, Petra, Richard, Shangsong, Tom, Wouter, Xinyi, and Zhaochun. Thanks for the chats, feedback, and social activities. Special thanks to Caroline and Petra for taking care of a lot of things. Chuan and Nikos, for helping with the final thesis production and the defense. Tom, for translating the thesis summary, and Mostafa, for designing the cover.

I want to thank my colleagues at KITLV for making me feel welcome at the institute. Being around so many experts on Indonesia was inspiring. Thanks for the support to our project, and for the fun outings and dinners. Special thanks to Ellen, Ine, and Yayah for helping me with all kinds of administrative bits over the years.

I was lucky to be part of a unique mix of people at the KNAW e-Humanities group, ran by Sally, Andrea, Jeannette, and Anja. It was exciting to see the different directions our projects took. Thanks for sharing the challenges of working in inter-disciplinary projects. Seeing how research is performed in other disciplines has truly been insightful.

Over the course of my PhD, I was fortunate to spend time in industry, on two occasions. These experiences have been very rewarding and have helped shape me as a researcher. Most of the work done during my internships has contributed directly to this thesis. First, I want to thank the members of the semantic search group at Yahoo Labs in Barcelona, where I learned a lot of useful things. Second, I want to thank everyone at Bloomberg's search team in New York, in particular Josh and Jon, for making the summer of 2016 a fun and productive one.

Being part of the Indonesian community in Amsterdam has definitely made this journey easier. I want to thank Adel, Arief, Damar, Hana, Haryo, Hendra, Heni, Laras, Marta, Pram, Said and everyone else in Science Park and PPI Amsterdam for the friendship, the support, and the fun occasional gatherings.

I want to thank my lecturers in Informatika ITB, in particular Bapak Dwi Hendratmo Widyantoro, Ibu Ayu Purwarianti, and Bapak Rila Mandala. Thanks for introducing me to the interesting worlds of information retrieval, natural language processing, and machine learning many years ago.

Finally, I want to thank my family for the unwavering support. Thanks, Uni and Ifa, for taking care of things while I'm away. Mama and Papa, thank you for letting me choose my own path in life.

Contents

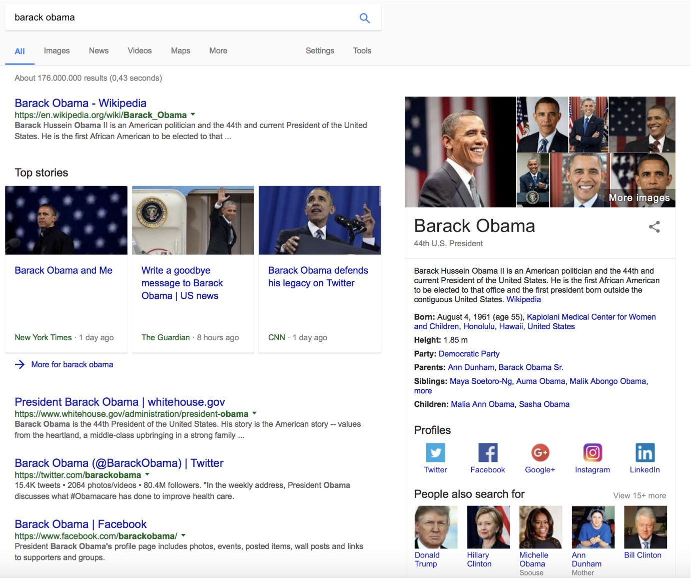
1 Introduction

Information retrieval makes dealing with a large amount of textual data manageable. Historians can select materials that are relevant to their research on a historical figure; a business analyst can track the ups and downs of a company. Such scenarios are often made possible by the following organizing units: *entities*.

Entities are commonly defined as things that have existence in the real-world, such as persons, organizations, and locations [138]. Being identifiable, independent objects, they make nice shortcuts to access information. To illustrate this in the context of Web search, nowadays it is estimated that as many as 71% of Web search queries contain entities [136]. In most common situations, users attempt to find information about an entity by issuing a query in the form of an entity (such as an entity plus a refiner, or the type of the entity).

An entity rarely stands on its own, however. Associations with other objects such as related entities, attributes, and topics are also of importance when working with an entity. For a user planning a vacation to a destination, for example, information related to the destination such as weather, travel, and attractions are also likely to be relevant. For a historian researching a particular figure, contextual information such as related persons and events will be of great importance. For a business analyst exploring investment opportunities, knowledge of supply chain and company products will be instrumental. One way to represent entity associations with other objects is through knowledge graphs [212]. Knowledge graphs typically encode information about entities with their types, attributes, and relationships in a graph format: nodes represent entities or entity types, edges represent relations.

Back to search. How can search systems benefit from understanding entity associations? Modern search engines typically perform more than just document retrieval. Multiple components are simultaneously shown in an aggregated search interface, e.g., direct answer snippets, knowledge cards, and related entities list (see Figure 1.1). To make this possible, more and more search engines make extensive use of entity-oriented information stored in knowledge graphs. The first, obvious use of entity associations information is when displaying a summary of an entity in the form of a knowledge card. To show the appropriate knowledge cards, queries which contain an entity need to be interpreted and linked to the correct entities [24, 169]. Structured information in knowledge graphs also allows search systems to provide direct answers by translating the query into a structured format through semantic parsing [18, 251]. To provide another example, having entity association information can also be used for recommendation purposes



About 176.000.000 results (0,43 seconds)

Barack Obama - Wikipedia
https://en.wikipedia.org/wiki/**Barack_Obama**
Barack Hussein **Obama** II is an American politician and the 44th and current President of the United States. He is the first African American to be elected to that ...

Top stories

Barack Obama and Me
New York Times · 1 day ago

Write a goodbye message to Barack Obama | US news
The Guardian · 8 hours ago

Barack Obama defends his legacy on Twitter
CNN · 1 day ago

More for barack obama

President Barack Obama | whitehouse.gov
https://www.whitehouse.gov/administration/president-**obama**
Barack Obama is the 44th President of the United States. His story is the American story -- values from the heartland, a middle-class upbringing in a strong family ...

Barack Obama (@BarackObama) | Twitter
https://twitter.com/**barackobama**
15.4K tweets • 2064 photos/videos • 80.4M followers. "In the weekly address, President **Obama** discusses what #Obamacare has done to improve health care.

Barack Obama | Facebook
https://www.facebook.com/**barackobama**/
President **Barack Obama's** profile page includes photos, events, posted items, wall posts and links to supporters and groups.

Barack Obama
44th U.S. President

Barack Hussein Obama II is an American politician and the 44th and current President of the United States. He is the first African American to be elected to that office and the first president born outside the contiguous United States. Wikipedia

Born: August 4, 1961 (age 55), Kapiolani Medical Center for Women and Children, Honolulu, Hawaii, United States

Height: 1.85 m

Party: Democratic Party

Parents: Ann Dunham, Barack Obama Sr.

Siblings: Maya Soetoro-Ng, Auma Obama, Malik Abongo Obama, more

Children: Malia Ann Obama, Sasha Obama

Profiles

People also search for

Figure 1.1: An aggregated result in response to the query "Barack Obama."

[20, 23, 255, 256]. Finally, knowledge graphs can be used to improve document retrieval through query expansion [55] or modeling [199].

In this thesis, we investigate the main theme of *entity associations* for search. In particular, we study three kinds of association: *entity-entity*, *entity-document*, and *entity-aspect* associations. Entity-entity associations are reflected through relations and all additional attributes of the relations, such as temporal boundary. Entity-document associations concern the relevance of a document with respect to an entity, and vice versa. Entity-aspect associations consider the relationship between an entity and related pieces of information attached to it. We touch upon various domains, starting with specific domains such as humanities and business, and ending in Web search. In addition, we consider search in a broad sense, exploring tasks beyond document retrieval such as object retrieval and ranking, recommendation, and filtering.

1.1 Research Outline and Questions

The main theme of this thesis concerns *entity associations*. We aim to address the following broad question: *how can we compute different types of entity association for search?* When addressing this question, we consider three types of associations for entities: *entity-entity*, *entity-document*, and *entity-aspect* associations. Individual work exploring each type of association exists, but a thorough investigation of all three themes has not been considered before.

Entity-entity associations

Our first theme is the association between an entity and other entities. In practice, such associations manifest themselves in relations between a pair of entities. The relation can be typed, i.e., it contains a specific semantic meaning, or not. In addition, these relations can have additional attributes detailing the nature of the relations. We explore the notions of non-typed, typed, and relation attributes in this first theme.

Driven by the need to support humanities researchers to explore large document collections, we set out on our first study. Increasing digitization and curation of humanities content in digital libraries gives rise to a new and interesting set of opportunities. New methods to enable these researchers to work on such large collections are needed [146]. We begin our investigation by considering entity networks as a means of exploration. The motivation is that providing an entity network would help users who are asking specific questions about the network or trying to discover interesting, non-obvious connections.

Drawing inspiration from earlier work on related entity finding and relation extraction, we formulate entity network extraction as the task of ranking related entities. Work on related entity finding aims to find related entities of a specific nature given a narrative of the expected relationships [36]. We rank related entities based on features derived from text, without specifying the relationships. In this setting, entity connections are formed only from their co-occurrence relationship in the text. In this first study, we ask the following question:

RQ1 How do we rank related entities to support the exploration of a document collection relying on signals from the text alone?

In our second study we go in a different direction. We focus on an important attribute of relations: *time*. Some entity relation types are *fluent*, i.e., they have a specific beginning and ending time [85, 112, 130]. Some examples of such fluent relation types are: *work-for*, *married-to*, and *attend-school* relations. Enriching these relations with their respective temporal boundaries can be important to support other tasks such as ranking related entities when we want to incorporate the temporal dimension.

We focus on establishing temporal boundaries between two entities having confirmed the relations between them. We assume entity relations are known, and the goal is to enrich the relations with their temporal boundaries. The extraction of temporal information is typically performed in three steps: (1) retrieval, detection and normalization of temporal expressions, (2) classification of each piece of evidence, and finally (3) aggregation of the evidence [112]. In this second study, we focus on the second step: temporal evidence classification. As the number of possible ways to express temporal relations is large, supervised approaches are ill-suited for this task. This challenge encourages us to turn to *distant supervision* [160, 205, 213]. Our second study is devoted to a specific problem with the distant supervision approach, captured by the following question:

RQ2 How can we effectively classify temporal evidence of entity relations?

In our last study on entity-entity associations, we return to ranking related entities in the context of recommendations. At this point, we consider the types of entity relations, assuming the existence of a knowledge graph with facts about entities and their connections. We study entity recommendations based on the notion of *impact*: tangible effect or consequence of an event involving a query entity to its related entities. In this task, we explore the business and political domains and work with knowledge graphs containing entities and relation types specific to these domains. In the political domain, for example, the use case that we consider involves estimating the vulnerability of political allies when a major event such as a corruption scandal strikes a politician.

Most work on entity recommendations is based on behavioral information in the Web domain [23, 119], or based on content such as the textual context in which the query entity appears [80, 127]. No previous work relies primarily on knowledge graphs and the semantics of the connections alone for recommendations. We ask how such recommendations can be generated assuming the existence of a knowledge graph. Knowledge graphs are heterogeneous in nature, containing multiple object and relationship types. Specifically in this study, we are particularly interested in the highly heterogeneous setting: graphs with a large number of relation types and object types. In a lower heterogeneity setting [126], it is feasible to learn the relative importance of each possible path directly. With highly-heterogenous graphs, the number of unique paths grows exponentially. This motivates us to find a solution that is able to model impact from a sequence of direct relations. In this setting, we are interested in the following question:

RQ3 Given graph-based information of entity relations with types, can we effectively recommend related entities based on their direct and indirect connections to a query entity?

Entity-document associations

We move on to our second theme: the associations between entities and documents in which the entities are mentioned. In practice, this type of association can be explored in both directions, i.e., translated as the relevance of a document given an entity, or the salience of an entity within a document. This type of association has been explored over the years, for example in the context of expert finding [11] or entity ranking for query understanding [200]. In both of these two scenarios, a ranking of documents mentioning a set entities is first performed, and entity-document associations are later used to obtain a ranking of these entities.

We study this type of association in the context of filtering documents for knowledge base acceleration, in which we need to filter documents relevant to update a profile of an entity, thus performing *entity-centric document filtering* [78]. In the context of search, this is related to a classic information retrieval task: *document filtering*. Document filtering aims to identify relevant documents given a dynamic, changing document collection, and a standing query [4].

Different approaches for entity-centric document filtering have emerged over the years, and they can be grouped into two main approaches: *entity-dependent* and *entity-independent*. In entity-dependent approaches, one model is learned for each entity, while in entity-independent approaches, a single model is learned to perform filtering for all entities. Extrinsic signals from documents, tweets, and queries are known to be very effective as they indicate an important event around entities [11]. For long-tail entities, however, these signals might not be as prominent, making us consider intrinsic, i.e., *in-document* signals more. We focus our effort on long-tail entities, and ask the following question:

RQ4 How do we filter documents that are relevant to update an entity profile, if the entity is in the long-tail?

Entity-aspect associations

Finally, we look into the associations between an entity and its aspects. Entities are often associated with attributes, types, distinguishing features, topics, or themes. We broadly group this type of information under the heading “aspect.”

In our last study, we study entity aspects in the context of Web search, and define them as common search tasks in the context of an entity. We focus on understanding aspects from query logs. We go beyond common knowledge graphs in which relation schemas are defined by a number of domain experts. Here we attempt to leverage users’ search queries to decide which entity-related information is valuable. Collecting these entity aspects is already valuable on its own, as it can help in determining the type of information for prioritizing entity relations to complete when building and updating knowledge graphs, or for designing presentation of entity profile in a search result.

Specifically, we study the problem of mining, ranking, and recommending aspects. The first challenge to address here is understanding different expression of queries that represent the same entity aspects. After that, we need to address the problem of estimating the importance of the aspect, and the connection between different entity aspects as they are asked by the users. We aim to answer the following question:

RQ5 How can we mine and represent common information needs around entities from user queries? How do we rank and recommend them?

We tackle the five research questions that we have listed above in five research chapters. Before moving on to the main contributions, in the next section we discuss other themes running in parallel with the overall theme of entity associations in this thesis.

1.2 Parallel Themes

In addition to the main theme of *entity associations*, the work presented in this thesis can also be examined from two other angles: (1) *enriching and leveraging entities*, and (2) *knowledge graph evolution*. In this section, we connect these two additional angles with the main theme that we have described earlier.

Concerning the first angle, *enriching entities* involves any attempt of extracting and obtaining more contextual information about entities, e.g., temporal relations, documents that are relevant to entities, and different aspects of entity-related information. As to *leveraging entities*, we use these enriched entities to improve information retrieval in general, through tasks such as ranking related entities, entity recommendation, and query recommendation. In some settings, i.e., RQ2, RQ4 we are mostly concerned with enriching entities, i.e., with temporal information and with new facts from a stream of documents. In the setting of RQ3 we specifically focus on leveraging entities; that is, assuming the existence of a knowledge graph, we generate recommendations from it. The bookending studies, i.e., RQ1 and RQ5 involve both enriching and leveraging entities, as we enrich entities with related entities and other aspects first, and also show how the enriched entities can be used to support exploration and query recommendation.

The concept of knowledge graph is pervasive in this thesis. We observe different states of knowledge graphs, starting from the simplest one, and ending with something that goes beyond current well-known knowledge graphs, motivating the idea of *knowledge graph evolution*. In the study of RQ1, we start with a fairly simple entity graph encoding co-occurrence relations in a document. This graph is homogeneous in nature as only co-occurrences between two entities are considered to connect two entities. In the setting of RQ2, we have a full knowledge graph with different relations and aim to enrich known relations with temporal information, attaching more information to the edges in the graphs. In the setting of RQ3, we consider a knowledge graph with a large number of relations, working with a highly-heterogeneous graph for entity recommendations. When investigating RQ4, we step back and consider the task of filtering documents that can be used to construct a knowledge graph from scratch or to update an existing knowledge graphs with new information. Finally, in the setting of RQ5, we move beyond the current notion of knowledge graph and enrich entities based on common information that users ask, mined from query logs.

1.3 Main Contributions

In this section, we summarize the main theoretical, algorithmic and empirical contributions of this thesis.

Theoretical contributions

Entity network extraction We describe how researchers can explore document collection through entity network extraction, formulated as the task of ranking related entities.

Impact-based entity recommendation We introduce a novel notion of entity relatedness: *impact*, and formalize the task of impact-based entity recommendations from knowledge graphs.

Mining, ranking and recommending entity aspects We introduce the notion of *entity aspects* and formalize three related tasks around it: mining, ranking and recommending entity aspects.

Algorithmic contributions

Method for entity network extraction We propose a method to rank related entities from text features based on statistical associations and linguistic features derived using relation extraction strategies to support exploration of a document collection.

Method for temporal evidence classification We propose a method for temporal evidence classification based on distant supervision that matches the distribution of source (i.e., distant supervision) corpus and target corpus when generating training examples automatically.

Method for impact-based entity recommendations We propose two methods for impact-based entity recommendations from knowledge graphs. We propose a probabilistic model to perform sequential prediction on knowledge graphs, and a learning to rank method based on features of query subgraphs.

Method for document filtering for long-tail entities We propose a method for document filtering that is tailored to improving performance on long-tail entities. We define three key intuitions concerning important documents in the document filtering setting and formulate these notions as a document representation for filtering.

Methods for mining, ranking and recommending entity aspects We propose methods for mining entity aspects from query logs by first performing entity linking on the queries and grouping the query context terms. We collect these aspects from query logs and propose methods for ranking and recommending them based on behavioral features extracted from the query logs, and the semantic relatedness of the context terms.

Empirical contributions

Entity network extraction We show that statistical association and relation extraction can be combined to improve performance when ranking related entities given a document collection. We identify common errors made with a co-occurrence approach.

Temporal evidence classification We show that rebalancing class labels to an empirical distribution can help improve the performance of distant supervision approaches. We compare the effectiveness of different learning algorithms and sentence representations in temporal evidence classification.

Impact-based entity recommendations We show how our proposed models can make predictions in an efficient way while obtaining an improvement over a strong graph proximity-based baseline. We learn that the learning to rank and graph-based approaches are complementary in providing quality recommendations on the task of impact-based recommendations. In addition, we learn that a graph-based approach produces better recommendations when faced with non-trivial entity candidate subgraphs.

Document filtering for long-tail entities We show that we can tailor document filtering methods towards long-tail entities without sacrificing overall performance. We compare the performance of our method on different query segments and identify important features and representations that work particularly well in this setting.

Mining, ranking and recommending entity aspects We show that a combination of lexical and semantic features is useful for grouping context terms required when mining entity aspects. We learn that entropy-based methods are suitable to rank entity aspects. As to aspect recommendation, we show that behavioral methods are superior to a semantic approach, but having a semantic approach can complement the behavioral approach in the event of sparsity.

1.4 Thesis Overview

This thesis is organized into eight chapters. After a background chapter in the form of a survey on related research areas, we continue with five research chapters and end with a concluding chapter. Due to the diverse nature of the topics covered in this thesis, additional in-depth background material and methodologies specific to each topic will be introduced as required. We present the following chapters:

Chapter 2—Background Here, we start with an in-depth survey on knowledge graphs from an information retrieval perspective. We examine information access tasks related to knowledge graphs from two main directions: how information retrieval techniques can be used to construct and enrich knowledge graphs, and how knowledge graphs can be leveraged to improve information retrieval.

Chapter 3—Entity network extraction We introduce the task of entity network extraction, formulating this task as the problem of *ranking related entities* to support the exploration of document collections. As the goal is to infer connections primarily from the text without specifying the type or nature of each relation, we rely on ranking related entities based on association measures extracted from text and features inspired from relation extraction approaches.

Chapter 4—Temporal evidence classification We focus on the specific problem of annotating relations between two entities with *temporal boundary*. More specifically, we focus on the task of *temporal evidence classification*: classifying a temporal expression in a sentence mentioning entity relations. We present a distant supervision approach to solving the task, which attempts to match the distributions of the target and distant supervision corpus.

Chapter 5—Impact-based entity recommendations from knowledge graphs Here we continue the theme of the first research chapter: *ranking related entities*, albeit in a different setting. In this chapter, instead of text a knowledge graph with multiple relation types connecting entity pairs are now given. We consider a novel notion of relatedness: *impact*, and formalize the task of *impact-based entity recommendations*. In this task, the goal is to predict the propagation of impact given a query entity of connected entities in a knowledge graph and rank the connected entities based on the predicted impact for recommendations.

Chapter 6—Document filtering for long-tail entities We consider the task of document filtering for entities in the context of knowledge base acceleration. Specifically, we focus on *long-tail entities* for which fewer signals are available to perform filtering. We propose and investigate several intrinsic features aimed to capture the importance of a document, inspired by the notions of *informativeness*, *entity-salience*, and *timeliness*. We propose a model that is *entity-independent*: learning the characteristics of an important document without the specific descriptions of each entity.

Chapter 7—Mining, ranking and recommending entity aspects We introduce and investigate the notion of *entity aspect*, defined as common search tasks in the context of an entity. We study three tasks: mining aspects of an entity, ranking aspects by importance to an entity, and recommending other, related aspects given an input aspect. We propose several methods for each task and perform experiments based on a commercial search engine query logs.

Chapter 8—Conclusions We conclude by summarizing our main findings and pointing out directions for future research.

1.5 Origins

This thesis is based on six publications in total. Here we list the original publications on which the background and research chapters are based, including the role of each co-author.

Chapter 2 This chapter is based on Reinanda, Meij, and de Rijke [188] "Knowledge Graphs: An Information Retrieval Perspective," submitted to a journal, 2017. Reinanda wrote an initial draft of the survey. All authors contributed to the organization of the material, the synthesis, and the writing of the text.

Chapter 3 This chapter is based on Reinanda, Utama, Steijlen and de Rijke [185] "Entity Network Extraction based on Association Finding and Relation Extraction," published at TPDL 2013. The design of the algorithm and experimental setup was due to Reinanda and de Rijke. The experiments were carried out by Reinanda. Utama and Steijlen mainly contributed to the construction of the dataset. All authors contributed to the text.

Chapter 4 This chapter is based on Reinanda and de Rijke [184] "Prior-informed Distant Supervision for Temporal Evidence Classification," published at COLING 2014. The design of the algorithm and experiments was mostly due to Reinanda. Both authors contributed to the text.

Chapter 5 This chapter is based on Reinanda, Pantony, Meij, and Dorando [189] "Impact-based Entity Recommendations from Knowledge Graph," submitted to KDD 2017. Reinanda led the development of the algorithms with contributions from the co-authors. Reinanda carried out the experiments and the analysis. All authors contributed to the text.

Chapter 6 This chapter is based on Reinanda, Meij, and de Rijke [187] "Document Filtering for Long-tail Entities," published at CIKM 2016. The design of the algorithms and experiments was mostly due to Reinanda. All authors contributed to the text.

Chapter 7 This chapter is based on Reinanda, Meij, and de Rijke [186] "Mining, Ranking and Recommending Entity Aspects," published at SIGIR 2015. All authors contributed to the design of the algorithms and experiments. Reinanda performed most of the experiments with contributions from Meij. All authors contributed to the text.

Finally, work on other publications also contributed indirectly to this thesis. Some of these work involve multi-disciplinary collaborations:

- Cai, Reinanda, and de Rijke [42]. "Diversifying Query Auto-completions." *ACM Transactions of Information Systems*, 2016.
- Hicks, Traag, and Reinanda [101]. "Turning Digitised Newspapers into Networks of Political Elites." *Asian Journal of Social Science*, 2015.
- Hicks, Traag, and Reinanda [100]. "Old Questions, New Techniques: A Research Note on the Computational Identification of Political Elites." *Comparative Sociology*, 2015.
- Reinanda, Odijk, and de Rijke [185]. "Exploring entity associations over time." *Proceedings of Time-Aware Information Access workshop*, 2013.
- Traag, Reinanda, and van Klinken [228]. "Structure of a Media Co-occurrence Network." *Proceedings of European Conference on Complex Systems*, 2014.
- Traag, Reinanda, and van Klinken [227]. "Elite Co-occurrence in the Media." *Asian Journal of Social Science*, 2015.

2
Knowledge Graphs: An Information Retrieval Perspective

In this chapter, we discuss related work on knowledge graphs from an information retrieval perspective, surveying various tasks and approaches in this area. We identify different sets of tasks related to knowledge graphs and information retrieval and group individual tasks that are closely related. As our main organizing principle, we grouped the tasks in two directions: *information retrieval for knowledge graphs* and *knowledge graphs for information retrieval*.

We first introduce and standardize the terminology that will be used for the rest of this chapter. As the work in this thesis is at the intersection of various fields, different terminologies with different usage appear. Our discussion rests on the following definitions. These definitions are partially based on [124, 153, 202].

Definition 1 (Entity) *An entity is an atomic, identifiable object which can have an a distinct and independent existence.*

Definition 2 (Mention) *A mention is a text segment which refers to an entity.*

Definition 3 (Relation) *A relation is an instance of relationship between two entities, of which the nature of the relationships can be defined with a label.*

Definition 4 (Attribute) *An attribute is a specific characteristic of the entity which has one or more values.*

Definition 5 (Knowledge base) *A knowledge base is a repository of entities with information about their relationships and attributes in a structured or semi-structured format.*

Definition 6 (Knowledge graph) *A knowledge graph is a knowledge base represented as a graph, specifically. In a knowledge graph, entities, attributes, relations are represented through the nodes and edges in the graph.*

Definition 7 (Entity profile) *An entity profile is a textual description of an entity in a knowledge base.*

2.1 Information Retrieval for Knowledge Graphs

In this section we discuss how information retrieval approaches can be used for improving and updating knowledge graphs. Table 2.1 presents a brief overview of the general taxonomy of tasks and approaches. We discuss each task in detail in the following subsections.

We start our discussion with a fundamental entity-oriented task: *entity recognition and classfication* (Section 2.1.1). Then, we touch on the issue of knowledge graph construction and completion, starting with discovering entities (Section 2.1.2), filtering relevant documents for entities (Section 2.1.3), and extracting relations between entities (Section 2.1.4). Finally, we discuss paradigms and approaches for estimating a knowledge graph's quality (Section 2.1.5).

Table 2.1: Taxonomy of tasks and approaches.

Task and approaches	**Description**
Entity recognition and classification (Section 2.1.1)	Detect segments of entity mentions within a text and the entity type.
rule-based	Handmade rules based on grammatical, syntactic, ortographic features, with dictionaries.
feature-based	Supervised and semi-supervised learning based on grammatical, syntactic, ortographic features.
embedding-based	Learn and associate mention, context and type labels in the embedding space.
Entity classification (Section 2.1.1)	Given an entity mention within a context, decide whether it belongs to a type.
feature-based	Supervised learning based on token, syntactic, ortographic, unigram and bigram features.
embedding-based	Learn and associate mention, context and type labels in the embedding space.
extractor-based	Extract type candidates based on patterns, mention text, and verbal phrases, rank semi-supervised fashion.

Entity discovery (Section 2.1.2)	Decide whether a new entity should be added as a new entry to a knowledge base.
linking-based	Utilize the confidence score from an entity linking system to detect unlinkable entities.
feature-based	Train classifier based on features from timestamp and text features in a supervised on semi-supervised fashion.
expansion-based	Discover new entities similar to a number seed entities.
Entity typing (Section 2.1.2)	Decide whether a type should be assigned to annotate an entity.
constraint-based	Define a set of class constraints and optimize through linear programming.
embedding-based	Learn the association between an entity and type embedding.
graph-based	Represent entities' associations with other entities, type context descriptions, and entity descriptions as a graph and optimize.
generative	Build co-occurrence dictionary of entities and context nouns, learn translation and generation probabilities.
Document filtering (Section 2.1.3)	Decide whether a document contains important information about an entity.
entity-dependent	Learn a model for every entity based on lexical and distributional features.
entity-independent	Learn a single model for all entities based on distributional features.
Relation extraction (Section 2.1.4)	Extract entity relations from text.
feature-based	Extract feature based on relation context within sentences, learn in a supervised fashion.

kernel-based	Design kernel functions to compare object structure similarity between training and test examples.
feature-based distant	Extract features based on relation context aggregated from multiple sentences, learn in distanly-supervised fashion.
pattern-based extraction	Learn an apply relation pattern in a semi-supervised fashion.
open information extraction	Extract relation-like facts without speficying a schema.
Link prediction (Section 2.1.4)	Predict new entity relations given known relations.
latent feature models	Learn latent features of entities that explain observable facts and apply to new entities.
graph feature models	Predict existence of new edge by learning features from the observed edges in the graph.
Triple correctness prediction (Section 2.1.5)	Estimate the likelihood of an entity relation triple.
fusion-based	Predict triple correctness by aggregating predictions of individual extractors.
Contribution quality estimation (Section 2.1.5)	Predict the quality of a knowledge base item (e.g., article).
feature-based	Predict contribution quality based on user contribution history, relation difficulty and user contribution expertise.
graph-based	Use the profile-editor and editor-editor graph structure to estimate the quality of contributed text.
Vandalism detection (Section 2.1.5)	Predict whether an edit in a knowledge base is malicious.
feature-based	Predict vandalism based on content and context features.

2.1.1 Entity recognition and classification

Recognizing entities in text is a well-known problem and one of the most fundamental entity-oriented tasks. The MUC-6 task [90] introduced the named entity extraction task and, later on, the CoNLL [222] and ACE evaluation campaigns [58] further drove research in this area. The task of named entity recognition is formally defined as follows:

Definition 8 (Entity recognition) *Given a piece of text s, detect segments of entity mentions m within the text.*

After the recognition phase, the type of an entity can also be detected. We define this entity classification task as follows:

Definition 9 (Entity classification) *Given an entity mention m within a context s, decide whether m belongs to a type $t \in T$, where T is a type classification system.*

Initially, the research in this area focused on classifying entities to broad classes such as *person*, *organization*, *location*, etc. Later on, more fine-grained class hierarchies were proposed, for instance by Sekine and Nobata [203].

Approaches

We discuss approaches to entity recognition and classification below. First, we discuss approaches that solve both recognition and classification, and then we continue with approaches that focus on the classification step only.

Entity recognition and classification The following approaches attempt to solve entity recognition and classification jointly.

Rule-based approaches Early approaches to named entity recognition and classification rely on dictionaries and handcrafted rules. A typical entity recognition rule could utilize the following signals: literal string, word class, part-of-speech, and previous named entity tagging label. A complex named entity tagger can be built by formulating and combining sets of these rules [203].

Feature-based approaches Rather than specifying complex rules manually, supervised learning approaches aim to learn to classify entities from data using contextual clues similar to the rule-based approaches. Supervised learning approaches to entity recognition utilize different classes of learning algorithms such as Hidden Markov Models [21], Decision Trees [201], Maximum Entropy [33], Support Vector Machines [6], and Conditional Random Fields [149]. The problem is often formulated as a sequential classification problem: tagging words within a sentence sequentially to indicate whether they are a part of a named entity or not.

Feature-based approaches can also be trained in a semi-supervised fashion; starting with a small number of seeds, building contextual clues relevant to these seeds, and then generalizing the pattern to recognize new entities. Approaches belonging to this category are presented in [51, 53, 172, 196].

Embedding-based approaches Ren et al. [190] propose a joint approach to entity recognition and classification based on distant supervision. They perform phrase mining to generate entity mention candidates and relation phrases and enforce the principle that relation phrases should be softly clustered when propagating type information between their argument entities. The type of each mention is predicted based on the type signatures of its co-occurring relation phrases and the type indicators of its surface name. They formulate the joint optimization problem for the type propagation and relation phrases clustering tasks. This approach outperforms Stanford NER [74] on New York Times and

Yelp corpora, achieving F_1 scores of 0.94 and 0.79, respectively. On a Twitter corpus, it achieves lower precision than Stanford NER, with higher recall.

Entity classification Approaches that focus on entity classification are either *feature-based*, *embedding-based*, or *extractor-based*.

Feature-based approaches Ling and Weld [138] introduce a feature-based approach for fine-grained entity recognition based on multi-label classification. They employ features such as tokens, word shape, parts-of-speech tags, unigrams, or bigrams with multi-label classifiers based on perceptron. All non-zero prediction scores are considered as relevant types for entity mention m within a context s. The classifiers are trained with automatically generated training data. To generate this data, they utilize linked segments m_e in a sentence contained in the corresponding Wikipedia page for entity e, and retrieve the type from Freebase. Some heuristics are employed to remove sentences that might not be useful for training. When evaluated on a Wikipedia corpus, this approach outperforms Stanford NER by 11%. In addition, it was shown that incorporating type information can help improve the performance of relation extraction systems (Section 2.1.4).

Embedding-based approaches Dong et al. [59] introduce a hybrid neural model that classifies entity mentions into a set of entity types derived from DBpedia. They introduce a mention model to obtain the vector representation of an entity mention from the words it contains. Another component, the context model, obtains the representation of the contextual information around a mention. They utilize representations obtained from the two components to predict the type distribution. The two representations are learned from automatically generated training data based on linked entity mentions in Wikipedia, similar to [138].

The mention model is built on Reccurrent Neural Networks (RNN). The vector of an entity mention is computed from the vectors of the words in the mention. The goal is to learn a global composition function and word embeddings from data. The representation of phrase $w_1 w_2$ is computed as follows:

$$p = f\left(W \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + b^m\right), \tag{2.1}$$

where $w_1, w_2 \in R^{d\times 1}$ are single-dimensional word vectors, W the composition matrix, b^m the bias vector, and f a nonlinear function. The context model is based on a Multi-Layer Perceptron (MLP). Context words are represented as low-dimensional vectors that are different from the ones in the mention model. Context word vectors are concatenated and fed into a hidden layer which produces a l dimensional vector. We can compute the output of the hidden layer as:

$$h = f(H\left[c_1, \ldots, c_s\right] + b^c), \tag{2.2}$$

where c_i are the word vectors, $H \in R^{L\times DS}$ is the weight matrix, b^c the bias vector and f a non-linearity function.

The mention model and context model are jointly trained; they are both fed to a softmax classifier that will compute type assignment distributions. During training, the cross-entropy errors between the predicted and ground truth distributions are minimized, and the errors are backpropagated to the two models. This neural model outperforms a strong feature-based approach [138], obtaining a 2.5% improvement without hand-crafted features and external resources.

In contrast with the previous approach, Yogatama et al. [253] learn both instance feature vectors and type labels to a low dimensional space $\mathcal{R}^d$ in d dimensions such that the instance is close to its label in this space. Their approaches are based on type embeddings that allow for information sharing among related labels. The score of an entity type t and a feature vector x is obtained from the dot product of their embeddings:

$$s(m, y_t; \theta_x, \theta_t) = f(m, \theta_m) \cdot g(y_t, \theta_t) = \theta_m m \cdot \theta_t y_t, \tag{2.3}$$

where θ_x and θ_t are the network parameters, and f and g the mapping functions. In terms of performance, this method also significantly outperforms [138], further confirming the potential of neural methods for entity classification.

On a similar note, Ren et al. [191] propose an approach that is based on extracting text features for entity mentions, performing joint embedding of entity mentions M and type hierarchy T into the same low-dimensional space such that close objects also share similar types. They estimate the type-path $t*$ on the hierarchy T using the learned embeddings. This search is performed in a top-down manner, selecting the most similar types based on embeddings at every step. They introduce a novel embedding method to separately model clean and noisy mentions and incorporate a given type hierarchy to induce loss functions. They formulate a joint optimization problem to learn embeddings for mentions and type-paths and develop an iterative algorithm to solve the problem. This method turns out to be very successful, outperforming many feature-based and neural methods for entity classification, including [59, 138, 253, 254].

Fine-grained entity typing systems are typically trained in a distant supervision manner, utilizing labels from knowledge bases which might be incorrect in the local context for some mention. Following-up on this problem, Ren et al. [192] perform automatic identification of correct type labels for training examples, given the set of candidate type labels from a type hierarchy. This noise reduction strategy is very effective, improving the performance of [138] by up to 33.53%.

Extractor-based approach Extractor based-approaches are similar to feature-based approaches, but they specifically limit the possible type assignments by applying a set of extractors leveraging the following signals: patterns of explicit type mentions, specific prefix or suffix of a mention, verbs following an entity mention, and types of entities occurring in the similar context. Corro et al. [52] introduce a system that generates candidate types using a sequence of multiple extractors, ranging from explicitly mentioned types to implicit types, and subsequently selects the most appropriate type using ideas from word-sense disambiguation. It does not rely on supervision in its extractor and generates training data for type selection from WordNet, SemCor, and Ontonotes. Their approach, FINET, first generates a set of candidate types using multiple extractors based on patterns, mention text, verbal phrases, and related entities. After the candidates have been generated, they select the most appropriate type with a Naive Bayes classifier utilizing context features such as words in the sentence. They utilize WordNet to extract the context features based on the type's gloss and its neighbors' glosses, their neighbors and corresponding verbs. They later trained one classifier per coarse-grained types. Corro et al. [52] show that FINET tends to be precision-oriented due to its conservative nature of suggesting types. In addition, its performance is superior to that of a strong feature-based baseline, HYENA [254].

Relation to other tasks

Entity recognition and classification are fundamental tasks that enable many other downstream tasks. Multi-word expressions recognized as entity mentions can be used as candidates for *relation extraction* [7] (see Section 2.1.4). The entity type detected in the classification is an important feature for relation extraction systems [160]. The recognized mentions can be used as candidates for *entity linking* (Section 2.2.1). In *document filtering*, some important features are extracted by first the detecting mentions of entities in the document (see Section 2.1.3).

Outlook

Although entity recognition and classification performance for English have achieved a good performance for popular domains like news, achieving an F_1 score of 90.90 on the CoNLL 2003 test set [173], this performance does not always translate to all domains. Interesting research directions include *domain-specific entity recognition* and *entity recognition on lesser-resourced languages*.

Recent work towards the first direction exist. Prokofyev et al. [179] consider the task of named entity recognition for idiosyncratic document collections. Tao et al. [221] focus on entity extraction in an enterprise setting, while Tang et al. [220] consider the task of entity recognition and linking in a social media context. To improve the recognition performance on a specific domain, encoding more background knowledge in the recognition and classification algorithm is an interesting challenge.

For lesser-resourced languages, it would be interesting to apply transfer learning or distant supervision approaches to improve the entity recognition. One way to achieve this is by applying machine translation or a heuristic text alignment technique to generate pseudo-training data for the lesser-resourced language.

2.1.2 Entity discovery and typing

The set of entities in a knowledge graph tends to evolve as new entities emerge over time. To keep up with real world entities, we need to continuously discover emerging entities in news and other Web streams [89]. Entity discovery originated as a subtask of TAC-KBP, an evaluation campaign on entity linking and relation extraction [65]. We define the task as follows:

Definition 10 (Entity discovery) *Given a stream of documents S and an entity e, decide whether e is a new entity that should be added as a new entry to a knowledge graph.*

Related to the problem of discovering new entities is deciding which type(s) these entities should belong to. In constrast with the entity clasisification problem which decides entity types on the mention level, in entity typing we decide the type(s) of an entity based on pieces of evidence in a corpus. We formally define the task as follows:

Definition 11 (Entity typing) *Given a set of documents D_e mentioning an entity e, decide whether type $t \in T$ should be assigned to annotate entity e, where T is a type system in the knowledge graph.*

Approaches

Below we group approaches to entity discovery and typing.

Entity discovery

Linking-based approaches Originally, approaches such as [125] and [182] utilize a global threshold to recognize entities not found in the knowledge base by an entity linking method. Entity linking systems, which attempt to link entity mentions in a text segment to knowledge base entities, often generate confidence scores of the linking proces. One of the early approaches to entity discovery extracts candidates of emerging entities from unlinkable entities (the ones with low confidence scores).

Feature-based approaches Lin et al. [135] attempt to solve the problem of detecting new entities based on their usage characteristics in a corpus over time. Their intuition is that entities have different usage-over-time characteristics than non-entities. They define an entity as a noun phrase that could have a Wikipedia-style article if there were no notability or newness considerations. They address the task by training a classifier with features primarily derived from a time-stamped corpus. Various statistics of entity usage from a longitudinal corpus are extracted. Specifically, they utilize things such as slope, R^2 fit, and occurrence histories. In addition, word features of the noun phrases (e.g., capitalization, numeric modifier) are also incorporated. To evaluate the approach, two annotators labeled 250 unlinked bigrams as *entity*, *non-entity*, or *unclear*. The proposed approach with all features manages to classify 78.4% of the bigrams correctly, outperforming a named entity recognition baseline and the proposed approach with individual features.

Graus et al. [87] present a distant supervision method for generating pseudo-training data for recognizing new entities that will become concepts in a knowledge base. The focus is in the setting of social streams, specifically. An entity linking system is applied to identify concepts in each sentence in a document. The sentence is then pooled as a candidate training example. A named entity recognition system is trained on this automatically generated ground truth data. They hypothesize that new entities that should be included in the knowledge base occur in similar contexts as current knowledge base entities. They rely on features such as number of mentions, number of urls, average token length, density, and length to sample tweets to be used for training. Their method achieves 45.99% precision and 29.69% recall when detecting new entities on a sample of Wikipedia articles.

Wu et al. [242] propose an approach that learns a novel entity classifier by modeling mention and entity representations into multiple feature spaces. They incorporate features based on contextual, topical, lexical, neural embeddings, and query spaces. Contextual features include supportive entities, alien entities, and dependent words. Semantic relatedness is computed by the relatedness of the entity in an embedding space. Within the query space, context words found in users' search history surrounding the entities are included. All of these features are combined to train a classifier based on gradient boosting trees. The proposed approach outperforms two strong baselines [103, 182] on the AIDA-EE dataset [103], achieving 98.31% precision and 73.27% recall for novel entity discovery.

Hoffart et al. [103] focus on the most difficult case where the names of the new entities are ambiguous. They propose a method to solve this problem by measuring the confidence of mapping an ambiguous mention to an existing entity, and a new model of representing a new entity with the same ambiguous name as a set of weighted keyphrases. They extract descriptive keyphrases of a candidate emerging entity and compute the model difference between existing and emerging entities. They later cluster different mentions with similar keyphrases as a new emerging entity. Their approach relies on extracting keyphrases related to existing entities and also candidate emerging entities. They retrieve the context for high-confidence mentions, and identify keyphrases in these sentences by extracting all sequences of predefined part-of-speech tag patterns, mainly extracting proper nouns and technical terms.

Expansion-based approaches Expansion-based approaches leverage the existing category and attribute information found in knowledge graphs and discover new entities with similar attributes.

Bing et al. [22] develop a framework for Wikipedia entity expansion and attribute extraction from the Web. That is, they discover new entities from the Web and extract the corresponding attributes of these new entities. They take existing entities from a particular Wikipedia category as seed input and explore their attribute infoboxes to obtain clusters for the discovery of more entities belonging to the same category. They also aim to find out the attribute value of these newly discovered entities. They develop a semi-supervised learning model with conditional random fields to deal with the issues of extraction learning and a limited number of labeled examples derived from the seed entities. They make use of a proximate record graph to guide the semi-supervised learning process. The graph captures alignment similarity among data records. Then, the semi-supervised learning process can leverage the unlabeled data in the record set by controlling the label regularization under the guidance of the proximate record graph.

Cao et al. [45] consider the task of target entity disambiguation: identifying target entities of the same domain. They propose a graph-based model to collectively identify target entities in short texts given a name list only. A large number of web queries are related to product entities. Radhakrishnan et al. [181] consider the task of modeling the evolution of product entities. More specifically, they tackle the problem of finding the previous version of a product entity. Given a repository of product entities, they first parse the product names using a CRF model. After grouping entities corresponding to a single product, they solve the problem of finding the previous version of any given particular version of the product, solved with Naive Bayes classifier. A common behavior of users is to compare among various comparable entities for decision making, e.g., comparing phones, cars, etc. Jiang et al. [115] specifically address the task of discovering which entities are generally comparable from the users' viewpoint in the open domain. They propose a model to learn an open-domain comparable entity graph from the users' search queries. Their approach firstly mines seed comparable entity pairs from search queries using predefined query patterns. Then, it discovers more entity pairs with a confidence-based classifier in a bootstrapping fashion.

Entity typing Approaches to entity typing are either *constraint-based*, *embedding-based*, *graph-based*, or *generative*.

Constraint-based approaches Nakashole et al. [163] consider the task of both discovering and semantically typing newly emerging out-of-KB entities. Their method is based on probabilistic models that feed weights into integer linear programs that leverage type signatures of relational phrases and type correlation or disjointness constraints. Their solution leverages a repository of relation patterns that are organized in a type signature taxonomy.

The candidate types to be assigned to an entity is determined based on the entity's co-ocurrence with a type relational patterns. Their method starts with generating a number of confidence-weighted candidate types for entity e:

$$typeConf(e,t) = \sum_{phrase_i} \sum_{p_j} sim(phrase_i, p_j) * \Gamma, \tag{2.4}$$

where t is the candidate type, e is the target entity, $phrase_i$ is a phrase that appears between entity e and candidate type t, and p_j is a signature associated to entity types. Γ is a constant determined by association between entity, type, and pattern.

Finally, the compatible subsets for an entity e is decided with an Integer Linear Programming (ILP) formulation. The constraint is that some types are mutually exclusive. More formally:

$$\max \sum_i T_i w_i, \tag{2.5}$$

constrained on:

$$\forall (t_i, t_j)_{disjoint} T_i + T_j \leq 1, \tag{2.6}$$

where $(t_i, t_j)_{disjoint}$ are pairs of disjoint entity types, and T_i a decision variable that indicates the assignment of the type t for an entity. Here, the goal of ILP is maximizing the weight so that known disjoint types do not get assigned together for the same entity. Types of emerging entities from news data are collected through crowdsourcing for evaluation. Their best method achieves 77%-88% precision for detecting the types of these news entities.

Similar to [163], Dalvi et al. [56] also present a method that employs class constraints imposed by the ontology. They consider two kinds of type constraints: *subset* and *mutual exclusion*. These constraints are finally incorporated within a Mixed-Integer program (MIP) approach to estimate type assignment subject to the previous two constraints.

Embedding-based approaches Embedding-based approaches to entity typing learn classifiers over sparse high-dimensional feature spaces that result from the conjunction of elementary features of the entity mention and its context. Yaghoobzadeh and Schutze [246] propose an embedding-based approach that combines a global model that scores based on aggregated contextual information of an entity, and a context model that first scores the individual occurrence of the entity and then aggregates the scores. In contrast with named entity recognition, which only looks at occurrence in a particular context, they aim to optimize both the corpus-level and context-level performance.

They contrast two models for this task: a *global model* that aggregates contextual information about an entity e from the corpus then performs classification for each possible type t. They also propose a *context model* that makes decision on each occurrence of entity e within a context whether e expresses type t or not. Their global model utilizes entity embedding $\vec{v}(e)$ and makes classes prediction based on a neural approach:

$$S_{GM}(e,t) = G_t(\tanh(W_{input}\vec{v}(e))), \tag{2.7}$$

where $W_{input} \in R^{(h \times d)}$ is the weight matrix from $\vec{v}(e) \in R^d$ to the hidden layer with size h, and d is the dimension of the entity embeddings. G_t is a logistic regression classifier that is applied to the hidden layer. The parameters are learned with Stochastic Gradient Descent. The context model is defined as follows:

$$S_{CM}(e, t) = g(U_{e,t}), \tag{2.8}$$

where $U_{e,t}$ is a set of local predictions on the contexts, and g is an aggregation function applied on the local predictions, e.g., the mean. The local prediction is also learned through a Multi Layer Perceptron:

$$S_c = G_t(\tanh(W_{input}\psi(c))), \tag{2.9}$$

where $\psi(c)$ is the feature vector of context c in the embedding layer, and W_{input} the parameters of the hidden layer. Evaluated on a subset of the ClueWeb12 corpus, this approach achieves an F_1 score of 0.545 for entity typing, obtaining a substantial improvement over [135]

Graph-based approaches This type of approach models the relationships between mentions, entities, and types. Mohapatra et al. [161] present a joint bootstrapping approach for entity linking and typing. More specifically, they present a bipartite graphical model for joint type-mention inference. They evaluate their approach by evaluating on entities that appear in Freebase. Their typing approach is based on building models of contexts referring to types.

Their approach relies on three signals: “entity neighborhood,” “language model,” and “neighborhood match with snippet.” Entity neighborhood leverages the direct or indirect information of type information from known parts of that knowledge graph; that is, inferring an entity’s type based on types of related entities. The language model utilizes mention contexts from Wikipedia annotated text. Finally, the last signal utilizes the linked related entities in context.

Their inference approach is based on a graph-based method with maximum a posteriori (MAP) labeling, a collective inference approach. They model the probability of a joint assignment of entity mention m to type T_e and entity e_m as:

$$\log P(t, e) = \alpha \sum_e \phi_e(t_e) + \beta \sum_m \phi_m(e_m) + \sum_{e,m} \gamma_{e,m}(t_e, e_m) - const, \tag{2.10}$$

where ϕ_e and ϕ_m are node log potentials, $\gamma_{e,m}$ are edge log potentials, and α and β parameters of the models. Node log potentials of each type are estimated from the entity neighborhood signal, while the node log potential of the entity neighborhood is based on the linked related entities signal. Edge log potential is estimated by the cosine similarity of the language model. This graph-based method achieves 80% accuracy for classifying entity types on the ClueWeb12 corpus.

Generative approaches Bast et al. [19] propose a method for assigning relevance scores for entity type assignments. Their method makes use of existing facts in a distantly supervised fashion. They generate pseudo training data by assigning entities with *only* the given type or any specialization of it and negative examples based on people that do not have that particular type in a knowledge base. With each entity (person in this case), they associate all words that cooccur with a linked mention of the person within the same semantic context. They define a semantic context as a subsequence of the sentences that expresses one fact from the sentence. They consider three algorithms: binary classification based on the associated context words, counting profession words, and a generative model similar to LDA or PLSI. The type distribution is later computed from the maximum likelihood estimate by applying an Expectation Maximization procedure to infer the latent variable.

Hovy et al. [108] present an approach to learn domain structure from unlabeled text. They first generate candidate classes from the data, and then utilize syntactic constructions, including: nominal modifiers, copula verbs, and appositions. They extract the entity and proper noun pairs and collect their counts over a corpus to derive probability distributions. In later work, Hovy [107] considers an unsupervised approach to learning interpretable, domain-specific entity types from unlabeled text. It assumes that any common noun in a domain can function as potential entity type, and uses those nouns as hidden variables in a Hidden Markov Model. During training, the co-occurrence of entities and common nouns are extracted from the data.

Relation to other tasks

As we have discussed earlier, entity discovery can be performed alongside *entity linking*, utilizing linking confidence scores. We will discuss entity linking in detail in Section 2.1.3. *Document-filtering* systems (Section 2.1.3) can be used to automatically build an initial profile for a new entity discovered through entity discovery. Document filtering approaches focused on the long-tail are especially useful in this case. Entity typing is related to fine-grained *entity classification* at the mention level. Just as one of the previous methods described, local mention classification can be incorporated to perform global decisions on entity typing. Finally, entity discovery can be considered as a form of *knowledge base completion*, as it complements a knowledge base by adding new entities.

Outlook

Below we discuss two types of future work related to entity discovery and typing: *automatically generating the description of newly discovered entities*, and *dealing with dynamic typing systems*.

Related to the task of discovering new entities is extracting their textual descriptions. Hoffart et al. [104] develop a simple approach that allows searching for descriptions of emerging entities in a user-friendly manner. They refine the method in [207], requiring the user to provide a minimal description of an entity, that consists of a name and initial keyphrases. Both approaches rely on having a human in the loop; it would be interesting to explore purely automatic approaches to address this problem.

Dalvi et al. [56] address the challenge of discovering new entity types with exploratory learning, which allows classification of datapoints to a new class not found in the training data. Their approach can be improved by learning the association between a newly discovered type and existing types. Another interesting direction is automatically labeling each new type.

2.1.3 Entity-centric document filtering

Document filtering has been a traditional task in TREC in the form of Topic Detection and Tracking (TDT). TDT constitutes a body of research and an evaluation paradigm that addresses event-based organization of broadcast news. The goal of TDT is to break the text down into individual news stories, to monitor the stories for events that have not been seen before and to gather stories into groups that each discuss a single news topic [4]. In constrast with ad-hoc search where the collection is static, in topic detection and tracking the queries are static while the document collection is dynamic and continously updated in a streaming fashion.

Entity-centric document filtering is the task of analyzing a stream of ordered documents and selecting those that are relevant to a specific set of entities. Introduced as the TREC KBA (Knowledge Base Acceleration) track [77], various approaches have been proposed to tackle this problem. The task is formally defined as follows:

Definition 12 (Entity-centric document filtering) *Given a stream of documents S and an entity e, decide whether a document $d \in S$ contains important information about e.*

Approaches

Entity-dependent approaches When TREC KBA was first held in 2012, most methods relied on *entity-dependent*, highly-supervised approaches utilizing related entities and bag of word features. Here, the training data is used to identify keywords and related entities, and classify the documents in the test data.

Liu et al. [141] present a related entity-based approach. They pool related entities from the profile page of a target entity and estimate the weight of each related entity with respect to the query entity. They then apply the weighted related entities to estimate confidence scores of streaming documents. This approach achieves an F_1 score of 0.277 on the TREC KBA 2013 dataset; the official name-matching baseline obtains the F_1 of 0.290 that year.

Efron et al. [64] introduce an approach based on sufficient queries, that is, high-quality boolean queries that can be deterministically applied during filtering. With this approach, no scoring is necessary since retrieval of entity-centric documents is purely based on these boolean queries. On the TREC KBA 2013 dataset, this sufficient query approach achieves an F_1 score of 0.316. Dietz and Dalton [57] also propose a query expansion based approach on relevant entities from the KB.

Wang et al. [236] propose a novel discriminative mixture model based on introducing a latent entity class layer to model the correlations between entities and latent entity classes. This latent entity class is inferred based on information from a Wikipedia profile and category. They achieve increased performance by inferring latent classes of entities and learning the appropriate feature weights for each latent class. More formally:

$$P(r, z|e, d; \alpha, \omega) = P(z|e; \alpha)P(r|e, d, z; \omega), \tag{2.11}$$

where $P(z|e;\alpha)$ is the probability of choosing the hidden entity class z given entity e with the parameter α and $P(r|e, d, z)$ the probability of the relevance of document d that is assigned to latent document class z with respect to entity e; ω is the set of combination parameters, i.e., the weight of the feature vector of the respective latent class z. Since the model includes latent classes, the parameters of the model are learned with the Expectation Maximization (EM) procedure. In addition to entity features, their approach also takes into account hidden class features. This approach achieves the state-of-the-art performance on the TREC KBA 2013 dataset, obtaining an F_1 score of 0.407.

Entity-independent-approaches Models that rely less on the specifics of each entity began to emerge later during the organization of the campaign. Balog et al. [16] propose one such *entity-independent* approach. They study two multi-step classification methods for the stream filtering task, contrasting two and three binary classification steps. Their models start with an entity identification component based on alternate names from Wikipedia. They introduce a set of features that became commonly used in the subsequent TREC KBA campaigns. Evaluated on the TREC KBA 2012 dataset, this entity-independent approach achieves an F_1 score of 0.360, which is on par with the best performing methods of that year. To gain more insights, Balog and Ramampiaro [11] perform an experimental comparison of classification and ranking approaches for this task. Their main finding is that ranking outperforms classification on all evaluation settings and metrics on the TREC KBA 2012 dataset. Their analysis reveals that a ranking-based approach has more potential for future improvements.

Along this line of work, Bonnefoy et al. [27] introduce weakly-supervised, entity-independent detection of the central documents in a stream. Zhou and Chang [264] study the problem of learning entity-centric document filtering based on a small number of training entities. They are particularly interested in the challenge of transferring keyword importance from training entities to entities in the test set. They propose novel meta-features to map keywords from different entities and contrast two different models: linear mapping and boost mapping.

Wang et al. [235] adopt the features introduced in [11] and introduce additional citation-based features, experimenting with different classification and ranking-based models. They achieve the best official performance for vital documents filtering in KBA 2013 with a classification-based approach, obtaining an F_1 score of 0.330.

In contrast with previous years, TREC KBA 2014 focuses on long-tail entities, and less than half of the entities in that test set have a Wikipedia profile [78]. In that year, Jiang and Lin [114] achieves the best performance (F_1 of 0.533) using an entity-dependent approach that uses time range, temporal, profession, and action pattern features. Another notable approach within that year summarizes all information known about an entity so far in a low-dimensional embedding [44].

Relation to other tasks

Document filtering is related to other tasks mentioned in this chapter, in particular *entity recognition* (see Section 2.1.1) and *relation extraction*, which we will discuss in the next section. Running relation extraction systems on a collection with a large number of documents can be very expensive computationally, which makes it difficult to apply on a Web scale. Document filtering selects a pool of documents for relation extraction. Document filtering can be used to help build an initial profile for entity discovery by selecting relevant documents in which the entity appears. Document filtering uses named entity recognition to extract entity features from the candidate document.

Outlook

Two future directions can be identified for entity-centric document filtering: improving the filtering performance on *long-tail entities*, and designing filtering approaches that can be applied to *unseen entities*. In this direction, Reinanda et al. [187] introduces a document filtering approach focused on long-tail entities (see Chapter 6). They introduce several intrinsic features which can be extracted only from the documents, and study how they learn a single, global model for entity-centric document filtering that can be applied to long-tail entities and entities not found in training data.

2.1.4 Relation extraction and link prediction

Relation extraction originated in the original information extraction tasks of slot filling that were first introduced in the Message Understanding Conference (MUC) series. The ACE evaluation campaigns [58] formally defined and included the task in 2002. To build a knowledge base from entity relations from scratch, relationships between entities must be extracted, a task that is commonly known as relation extraction in the natural language processing community. We define the task formally as follows:

Definition 13 (Relation extraction) *Given a sentence s containing a pair of entities e_1 and e_2, decide whether e_1 and e_2 are connected through a relation of type r.*

The incompleteness of knowledge graphs drives a lot of research in knowledge base completion, in particular link prediction. The link prediction task can be formally defined as follows:

Definition 14 (Link prediction) *Given a set of facts F about entity relations, predict the existence of new relations between to entities e_1 and e_2 within relation type r.*

Approaches

We briefly discuss approaches to relation extraction and continue with more detailed discussions on link prediction below. We refer to a survey on relation extraction by Bach and Badaskar [7] for a more comprehensive introduction on relation extraction. The discussion on link prediction methods is partially inspired by [166].

Relation extraction

Feature-based approaches Supervised methods for relation extraction are typically grouped into two classes: feature-based and kernel-based methods. In the feature-based methods, syntactic and semantic features are extracted from the text. Syntactic features often include the entities, the types of the entities, word sequences between the entities, and the number of words between the entities. Semantic features are derived from the path in the parse tree containing the two entities [92, 118, 260].

Kernel-based approaches To take advantage of information such as parse trees and to avoid generating features explicitly, kernel methods are introduced. Examples are presented in their original representation, and a function within the machine learning algorithm will compute the similarity between training examples within this rich representation. This rich representation can be in the form of a shallow parse tree or a dependency tree [38, 54, 257].

Distant, feature-based approaches Another way of dealing with generating training data for distant supervision is based on pseudo-training data. Mintz et al. [160] pioneered the work in this area. Later on, Alfonseca et al. [2], Hoffmann et al. [105], Riedel et al. [194], Surdeanu et al. [214], Yao et al. [250] further refine the model by relaxing the assumptions introduced in the original method. For example, Surdeanu et al. [214] achieve this by assuming that at least one distant-supervision training example is correctly labeled.

More recently, Zeng et al. [258] propose a novel distant supervision model that takes into account the uncertainty of instance labels during training. Their model also automatically learns relevant features, avoiding the necessity of feature engineering. They do so by adopting a convolutional neural network architecture with piecewise max pooling to automatically learn relevant features. Semantic features include the path between the two entities in the dependency parse.

Semi-supervised pattern extraction Since labeled data is expensive to create in large quantities, some groups have started to investigate bootstrapping/semi-supervised approaches [1, 35]. The main idea is to start with a small number of seed relation instances, learn a general textual pattern that will apply to these relations, and apply the newly discovered patterns to discover more relations. Later on, web-scale approaches are introduced in [67].

Open relation extraction In work by Banko et al. [17] relations are extracted without normalizing them to a specific schema. Relation-like tuples are extracted from text after learning how relations are typically expressed. Open relation extraction approaches are based on features such as the existence of verb and capitalizations of words.

Link prediction Approaches to link prediction are either based on modeling latent features or existing connections in graphs.

Latent feature models Latent feature models are related to *tensor factorization*. Factorization models learn a distributed representation for each entity and each relation, and make predictions by taking the inner products of the representations. Sutskever et al. [215] are the first to propose the latent factor model approach to learning entity representations. Their approach utilizes learning the lower-dimensional representation of an entity while taking into account relation types by applying Bayesian clustering factorization techniques. The distributed representation is learned for each argument of the relation. The probability of each relation triple (a, r, b) is computed as follows:

$$f(a,r,b) = P(T(a,r,b) = 1|\theta) = \frac{1}{1 + \exp(-a_L^T R b_R)}, \tag{2.12}$$

where $\theta = a_L, a_R, R$ are the collective parameters of the model; a_L,a_R the vectors of right and left arguments of the relation, respectively. R is the matrix. This model is a part of stochastic bloc model that clusters entities and relations using a non-parameteric process.

One of the simplest latent feature models is *bilinear model*. In [164, 165] RESCAL, which predicts triples through pairwise interactions of latent features was proposed. RESCAL works by modeling the score of a triple (a, r, b) as follows

$$f(a,r,b) = aTW_r b = \sum_{i=1}^{H_c} \sum_{j=1}^{H_c} w_{abr} a_i b_j, \tag{2.13}$$

where $W_k \in R^{H_e \times H_e}$ is a weight matrix whose entries w_{ijr} specify how much the latent features i and j interact in the r-th relation. It is a bilinear model that captures the interaction between two entity vectors using a multiplicative term. A key feature of this model is pairwise interaction. During training, both the latent representation of entities and how they interact are learned. The method introduced in [110] also belongs to this category. They focus on addressing the challenge of multi-relational data, in which multiple relations between entities may exist. This model also has a bilinear structure.

The following methods utilize *neural tensor networks*. Socher et al. [209] introduce an expressive neural tensor network suitable for reasoning over relationships between two entities. Although most of the work in this area represents entities as discrete atomic units or with a single entity vector representation, they show that performance can be improved when entities are represented as an average of their constituting word vectors. They also show that these entity vectors can be improved when initialized with vectors learned from unsupervised large corpora. In addition, their model can classify unseen relationships, extending their model from their previous work in [208]. In their model, each relation triple is described by a neural network and pairs of database entities which are given as input to the relation's model. The neural tensor network replaces a standard linear neural network layer with a bilinear tensor layer that directly relates the two entity vectors across multiple dimensions. The model computes a score of how likely it is that two entities are in a certain relationship. The prediction can be computed as follows:

$$f(a, b, r) = w_r^T g(a^T W^{[1:k]} b + V_R[ab] + b_R), \tag{2.14}$$

where g is a non-linearity, W the parameter containing pairwise interaction between two latent factors, V the interaction between two entities, and b_R a bias vector.

Some methods aim to learn *structured embeddings*. Bordes et al. [28] propose a model that learns to represent elements of any knowledge base into a relatively low dimensional embedding vector space. The embeddings are established by a neural network whose particular architecture allows one to integrate the original data structure within the learned representations. Specifically, the model learns one embedding for each entity and one operator for each relation. They show that using kernel density estimation in the embedding space allows us to estimate the probability density within that space so that the likelihood of a relation between entities can be quantified. This approach also basically combines a multi layer perceptron with bilinear models. Structured embedding is a variant of latent distance models, i.e., it computes the distance between entities to indicate relatedness:

$$f(a, b, r) = -\left|\left|W_r^L a - W_r^R b\right|\right|_1, \tag{2.15}$$

where $W_r = [W_r^L; W_r^R]$ is the parameter that transforms the latent feature representations of entities to model relationships for the relation r, and $|| \cdot ||$ a distance function.

Bordes et al. [30] propose SME, a semantic matching energy function that relies on a distributed representation of multi-relation data. The embedding is learned with a neural network whose particular architecture allows the integration of the original data structure. A semantic energy function is optimized to be lower for training examples than for other possible combinations of symbols. Instead of representing a relation type by a matrix, it is represented by a vector that shares the status and number of parameters with entities.

The following *translation models* are continuations of structured embeddings. The main feature of translation models is that the mapping between two entities is obtained by applying a relation vector, instead of matrix multiplication. Bordes et al. [29] also propose TransE, a method that models relationships by interpreting them as translations operating on the low-dimensional embeddings of entities. The intuition is that for two entities x and y, the embedding of entity x should be close to the embedding of entity y plus some vector that depends on the relationship between the two entities. It learns only one low-dimensional vector for each entity and each relationship. Their main motivation is that translations are the natural way of representing hierarchical relationships that are commonly found in knowledge bases. The likelihood of a triple is defined as:

$$f(a, b, r) = -d(a + r, b), \tag{2.16}$$

where a, b are the representations of the entities, r the latent representation of relations, and d a distance function that can be applied to these representations.

Later on, Wang et al. [238] propose TransH, an improvement of TransE that consider certain mapping properties of relations including reflexive, one-to-many, many-to-one, and many-to-many relations. In addition, they also propose a method to construct negative examples to reduce false negative labels in training.

While TransE and TransH put both entities and relations within the same semantic space, an entity may have multiple aspects and various relations may focus on different aspects of entities. Lin et al. [137] propose TransR, a method to build entity and relation embeddings in separate spaces and then build translations between projected entities. Recently, Yang et al. [248] show that existing models such as TransE and TransH can be generalized as learning entities as low-dimensional vectors, and relations as bilinear/linear mapping functions for these entities. Ji et al. [111] propose a method to model knowledge graphs. They define two vectors for each entity and relation. The first vector represents the meaning of an entity or a relation. The other vector represents a way to project an entity embedding into a relation vector spaces. This means that every entity-relation pair has a unique mapping matrix. Luo et al. [148] also consider the problem of embedding knowledge graphs into continuous vector spaces. Existing methods can only deal with explicit relationships within each triple (local connectivity patterns) while ignoring implicit relationships across different triples (i.e., indirect relationships through an intermediate node). They present a context-dependent KG embedding method that takes into account both types of connectivity patterns and obtains more accurate embeddings.

Graph-based models Rather than learning the features of each entity and their pairwise interactions, graph-based random walk models utilize observed features found in the existing connections. One such model is the Path Ranking Algorithm (PRA) [126] that performs random walks of bounded lengths to predict relations. PRA learns the likelihood of each relation path, combining a bounded number of adjacent relations.

PRA ranks an entity e with respect to a query entity q by the following scoring function:

$$s(e; \theta) = \sum_{p \in \mathcal{P}(q,l)} h_{q,e} \theta_p, \tag{2.17}$$

where h represents the number of subpaths connecting two entities, and θ the weight of the path learned from training data.

One advantage of random walk models compared to latent factor models is their computational simplicity, although they tend to have lower inference accuracy due to the sparsity of connections in the graph. Gardner et al. [84] aim to improve the effectiveness of PRA by enriching knowledge graphs with additional edges. These additional edges are labeled with latent features mined from a large dependency-parsed corpus of 500 million Web documents. This enrichment is important to successfully improve the performance of PRA. Kotnis et al. [123] propose a method for knowledge base completion using bridging entities. Previous work has enriched the graph with edges mined from a large text corpus while keeping the entities (i.e., nodes) fixed. They augment a KB graph not only with edges but also with bridging entities mined from web text corpus. PRA is then applied to perform KB inference over this augmented graph, which helps to discover more relations. Another work on improving the performance of random walk models by addressing the sparsity issue is introduced in Liu et al. [140]. They propose a hierarchical random-walk inference algorithm which addresses the main problem of random walk models. They assume that entity relations are semantically bidirectional and exploit the topology of relation-specific subgraphs. From these assumptions, Liu et al. [140] design a model that combines the global inference on an undirected knowledge graph with the local inference on relation-specific subgraphs.

Gardner and Mitchell [83] extend PRA in a different way than [84, 123, 140]. They propose a simpler random walk algorithm that generates feature matrices from subgraphs. This method is proven to be more expressive, allowing for much richer features than paths between two nodes in a graph.

Another simple random walk model that uses observed features is proposed in [225]. It shows the effectiveness of observed features in comparison to latent feature models for knowledge base completion. They show that the observed features model is most effective at capturing the information present for entity pairs with textual relations, and a combination of the two combines the strengths of both model types. They incorporate both observed features from knowledge graph and also textual evidence, similar to [195]. Later on, Toutanova et al. [226] propose a model that captures the compositional structure of textual relations and jointly optimizes entity, knowledge base, and textual relation representations. The proposed model significantly improves over a model that does not share parameters among textual relations with common sub-structure.

Relation to other tasks

Relation extraction uses entity recognition (see Section 2.1.1) for candidate selection and features extraction; the documents from which the relations will be extracted are annotated with *entity recognition*. Relation extraction typically will only run on a selected pool of documents. Therefore it requires a *document filtering* component to run on the initial corpus. The output of relation extraction needs to be estimated by *quality estimation* components, which we will discuss in Section 2.1.5.

Entity relations extracted by the relation extraction and link prediction components will be leveraged by *entity linking* (Section 2.2.1), *document retrieval* (Section 2.2.2), *entity retrieval* (Section 2.2.3), or *entity recommendation* (Section 2.2.4).

Outlook

Interesting research directions on relation extraction and link prediction include the following: *leveraging multiple sources for knowledge base completion* and *targeted knowledge base completion with a budget*. We briefly discuss these directions below.

On the first direction, Zhong et al. [263] study the problem of jointly embedding a knowledge graph and a text corpus. The key issue is the alignment model which makes sure that the vector of entities, relations, and words are in the same space. They propose an alignment model based on the text description of entities. A possible extension of this work would be incorporating semi-structured data (e.g., extracted open relations) in combination with existing facts and text data for knowledge base completion.

Targeted knowlege base completion aims to discover new facts on specific relation types or entities. In this context, West et al. [240] propose utilizing a question-answering inspired approach for performing targeted relation completion. Hegde [97] aim to overcome the challenge of knowledge graph sparsity by focusing the completion on a set of target entities only. A possible, interesting extension along this line of work is a targeted knowledge base completion system with a budget; i.e., a system that can automatically make the decision on which entities or relations to be targeted first given a limited resource/budget. Adapting techniques from reinforcement learning would be suitable in this setting.

2.1.5 Quality estimation and control

Quality estimation of knowledge graphs can be considered a new area that is developing. Little work has been devoted towards ensuring the quality of facts contained in knowledge graphs. Work on this field can be divided into three paradigms: *triple correctness estimation*, *contribution quality estimation*, and *vandalism detection*. Below we discuss approaches related to each paradigm.

Approaches

Triple correctness prediction This paradigm focuses on estimating the correctness of a knowledge base relation triple

Fusion-based approaches Dong et al. [60] estimate the probabilities of fact correctness from multiple sources. They combine confidence scores from several text-based extractors and prior knowledge estimated based on known facts from existing knowledge repositories. These scores are then fused through and converted into a probability with a technique called Platt scaling. Their approach utilizes multiple relation extractors based on the text, html structure, and microformat annotations on the Web. They fuse the output of this system with a graph-based prior inferred from the current state of the knowledge graph. They consider two methods to compute the graph-based priors: (1) Path Ranking Algorithm (PRA) [126], and (2) an embedding method based on Multi Layer Perceptron.

The multi layer perceptron model is obtained by first performing a low-rank decomposition of the knowledge graph represented as the tensor $E \times P \times E$, obtaining the embedding entity, relation, and the other entity in the lower dimensional representation.

The probability can be computed as follows:

$$P(G(s,p,o)=1)=\delta\Big(\sum_{k=1}^{K}u_{sk}w_{pk}v_{ok})\Big), \tag{2.18}$$

where k is the dimension of the embedding, and $\vec{u},\vec{w},\vec{v}$ represent the embeddings. The output of the extractors and priors are then combined as features in a feature vector. Then, the weight of each feature is learned through a classifier such as linear logistic regression, or ensemble decision trees. To evaluate the performance of individual and combined approaches, Dong et al. [60] first generate a set of *confident facts*, i.e., facts which have estimated probability of being true above 90%. Then, they sample a balanced number of triples per relation type from this set, and compare the triples against the triples in Freebase.

The following are follow-up work to [60]. Dong et al. [61] compare different methods of aggregating knowledge, inspired by data fusion approaches. An approach to finding the systematic error during data extraction was proposed in [237]. Finally, Dong et al. [62] propose a method to decompose errors made during the extraction process and factual errors in the web source. The extraction performance is also evaluated by comparing the extracted triples against Freebase.

Contribution quality estimation This paradigm focuses on estimating the quality of a contribution on knowledge graphs,

Feature-based approaches Tan et al. [218] present a method for automatically predicting the quality of contributions submitted to a knowledge base. The proposed method exploits a variety of signals, including the user's domain expertise and the historical accuracy rates of different types of facts; this enables the immediate verification of a contribution, significantly alleviating the need for post-submission human reviewing.

The following signals are considered for prediction:

- **User contribution history** These features are meant to capture a user's reputation based on previous user's contributions, such as the total number of prior contributions, total number of correct contributions, and fraction of correct contributions.
- **Triple features** Triple features are aimed to capture the relative difficulty of each relation. This difficulty is estimated from the historical deletion rate of that particular predicate.
- **User contribution expertise** Users expertise is estimated based on previous contributions. They consider three different concept space: LDA topics, taxonomy, and triple predicate.

Tan et al. [218] introduce a new evaluation metric: *relative error reduction* (RER). Their proposed approach achieves the RER of 60%, a substantial improvement over baselines based on the following strategy: majority, users' contribution history, and users' long term contribution quality.

In line with previous work, Flekova et al. [76] study the user-perceived quality of Wikipedia articles. They utilize the Wikipedia user feedback dataset, which contains 36 million Wikipedia article ratings contributed by ordinary Wikipedia users. The ratings incorporate the following quality dimensions: *completeness*, *well-writtenness*, *trustworthiness*, and *objectiveness*. They select a subset of biographical articles and perform classification experiments to predict their quality ratings along each of the dimensions, exploring multiple linguistic, surface and network properties of the rated articles.

Graph-based approaches Li et al. [131] consider the problem of automatically assessing Wikipedia article quality. They develop several models to rank articles by using the editing relations between articles and editors. They develop a basic quality model based on PageRank. They represent articles and editors as nodes connected by edges that represent editing relations. The articles will are ranked by node value. To take into account multiple editors, they incorporate contributions made to an article and utilize these as edge weights during the PageRank computation.

Vandalism detection This paradigm focuses on detecting intentional vandalism action on structured and semi-structured knowledge bases. Vandalism is defined as malicious insertion, replacement, or deletion of articles.

Feature-based approaches Heindorf et al. [98] introduce a corpus for vandalism detection on Wikidata and perform some initial analysis on vandalism on this particular corpus. Later on, Heindorf et al. [99] propose a set of features that exploit both content and context information. Content features include features at the character, word, sentence, and statement level. These include capitalizations, character repetitions, profane/offensive words, and changes of suspicious lengths. Context features include things as the user, item, and revision features.

Research on vandalism detection originally spawned on unstructured and semi-structured knowledge bases, e.g., Wikipedia. Potthast et al. [177] were the first to render vandalism detection as a machine learning task. They compile some features for detecting vandalism on Wikipedia. Their method works at the *edit* level, where each edit consists of two consecutive revisions of a document. Each edit is then represented as a feature vector in which the classifier is applied. Currently, all vandalism detection approaches are *feature-based*.

Relation to other tasks

As we have discussed earlier, quality control has direct connections to *relation extraction and link prediction* (Section 2.1.4). In these two tasks, we would like to estimate the probability of extracted triples to be true. The correctness of entity relations can be incorporated in *entity recommendation* systems (Section 2.2.4) or *entity retrieval* systems which use relations between entities as features (Section 2.2.3). Any approaches that use entity profiles, such as *document retrieval* (Section 2.2.2), can benefit from the quality estimated by the estimation methods.

Outlook

We expect quality estimation models that take into account *multiple sources*, e.g., both text and graphs with more complex features to appear in the future. Such models can incorporate features extracted from articles or triples with relationship information between contributors or items. *Alternative validation strategy* is also an interesting research direction. One such strategy is presented in [232], in which video games are used for validating and extending knowledge bases.

2.2 Knowledge Graphs for Information Retrieval

In this section, we discuss how knowledge graphs can be used to help address various modern information retrieval tasks. Table 2.2 presents a brief overview of the general taxonomy of tasks and approaches.

We start our discussion with a fundamental task: entity linking (Section 2.2.1). Next, we discuss how entities detected in queries and documents can be used to improve document retrieval (Section 2.2.2). After that, we focus on the task of retrieving entities with respect to a query to satisfy an information need (Section 2.2.3), and continue with recommending related entities given an query entity (Section 2.2.4). To close this chapter, we discuss an emerging task: explaining relationships between entities (Section 2.2.5).

2.2.1 Entity linking

The goal of entity linking is to enrich a text with links to encyclopedic knowledge. Entity linking has its roots in natural language processing (i.e., evolved from cross-document coreference resolution) and database (record linkage). The specific task of linking entities to Wikipedia entry was popularized in [155] and later developed as one of the main tasks in the TAC KBP evaluation campaign [65]. The task is formally defined as follows:

Definition 15 (Entity linking) *Given a piece of text t, detect segments of entity mentions m within the text, and link them to an entry e in a knowledge base KG.*

A way of detecting such segments is through entity recognition, which we discussed in Section 2.1.1.

Approaches

Approaches to entity linking can be categorized into *feature-based, graph-based, neural, and joint* approaches.

Feature-based approaches Early work on entity linking was based on individual and intuitive features. In the early papers in this area, Mihalcea and Csomai [155] introduce the notion of *keyphraseness*: the probability of a term to be selected as a keyword in a document. Medelyan et al. [150] propose the notion of *commonness*: the overall popularity of a candidate entity as a target given a mention. Milne and Witten [159] introduce the notion of *relatedness*: the semantic similarity of two entities obtained from their incoming and outgoing links to related entities. They were the first to propose a machine learning approach to entity linking by combining the commonness and relatedness features. More sophisticated feature-based models incorporate signals derived from various sources.

Maximizing the relatedness of relevant entities selected as a reference will minimize disambiguation errors. Ceccarelli et al. [46] address the problem of learning entity relatedness functions to improve entity linking. They formalize the problem of learning entity relatedness as learning a ranking function and show that their machine-learned function

Table 2.2: Taxonomy of tasks and approaches.

Task and approaches	**Description**
Entity linking	Link an entity mention in text to an entry in a knowledge base.
feature-based	Disambiguate entity mention with features from entity, mention, and context.
graph-based	Disambiguate entity with context and coherence features, i.e., linking of other entities.
neural	Learn representation of entity and context, compare the similarity when disambiguating entity mention.
joint modeling	Learn model to perform entity linking and another task jointly.
Document retrieval	Retrieve documents given a query.
query expansion	Expand queries and document with entities and learn the associations.
latent factor modeling	Model object as a latent space between query and document, incorporate latent space in retrieval.
entity-based language modeling	Incorporate term sequences marked as entities when building the language models of a query and a document.
Entity retrieval	Retrieve relevant entities given a query.
language modeling	Retrieve entities by matching query with entity descriptions or mentioning documents.
neural language modeling	Learn latent representation of query and entities, compare for retrieval.
multi-fielded representation	Represent an entity as a multi-fielded document, learn to rank on the multi-fielded document representation.
Entity recommendation	Recommend related entities given an entity and/or context.
heuristic	Estimate statistical association between entities from text.
behavioral	Recommend entities based on similar users' interest.
graph-based	Recommend entities based on the connetions in graph.
Relationship explanation	Explain the relationship between a pair of entities.
instance-based	Explain relations by selecting a set of key related entities.
description ranking	Generate candidate description from external text source, rank based on various features.

performs better than previously proposed relatedness functions. Finally, they show that improving this ranking-based relatedness function also improves the performance of state-of-the-art entity linking algorithms. Similar to previous work in [46], Charton et al. [47] aim to leverage mutual disambiguation for entity linking. This goes with the idea that entity linking should maximize the relatedness of the entities in the candidate set.

Another similar work along this line is presented in [91]. When performing entity linking in microblog posts, they leverage additional resources, in particular, extra posts. First, they expand the post context with similar posts, i.e., they construct a query with the given post and search for more posts. Disambiguation will benefit from the extra posts if they are related to the given post in context and contain additional signals for disambiguation.

Graph-based approaches Some approaches perform entity disambiguation collectively, optimizing the coherence between candidate entities. Hoffart et al. [102] combine three important intuitions: the prior probability of an entity, the similarity between the context of a mention and a candidate entity, and also the coherence among candidates entities for all mentions together. Other seminal work includes [73, 182].

Recently, Alhelbawy and Gaizauskas [3] attempt to solve the disambiguation problem with a graph-based approach. They perform the disambiguation collectively by representing candidate entities as nodes and associations between different candidates as edges between the nodes. They rank the nodes with PageRank and combine it with an initial confidence score for candidate selection. Also, Ganea et al. [81] introduce a probabilistic entity linking approach that disambiguates entity mentions collectively. Disambiguation is performed by considering both the prior of entity occurrences and local information captured from mentions and their surrounding context. They rely on loopy belief propagation to perform approximate inference. Their approach relies on three sources of information: probabilistic name to entity map derived from a large corpus of hyperlinks, pairwise co-occurrence estimated from a large corpus, and contextual entity words statistics.

Neural approaches Cai et al. [43] propose an entity disambiguation model based on deep neural networks. Instead of utilizing simple similarity measures and their disjoint combinations, they directly optimize the document and entity representations. Their approach utilizes auto-encoders to learn an initial document representation in an unsupervised manner (pre-training). This is later followed by a supervised training to make the representation closer to a given similarity measure.

Joint approaches Recently a suite of related work which aims to jointly perform entity recognition and linking has appeared. Cross-document coreference resolution is a task that is closely related to entity-linking. The goal in this task is to compute equivalence classes over textual mentions denoting the same entity in a document corpus, without explicitly linking them to a knowledge graph entry, in contrast to entity linking. Dutta and Weikum [63] try to jointly solve the problem of cross-document coreference resolution and entity linking. Their method is *unsupervised*, where the output of the coreference resolution and informs the entity linking, and vice versa. The coreference resolution and linking steps are performed alternately in an iterative fashion that focuses on the highest-confidence unresolved mentions. Sil and Yates [206] propose a re-ranking approach for joint entity recognition and linking. They propose a joint model for these tasks by retrieving a large set of candidates of entity recognition and linking, and later ranking pairs of candidate-entity mention. The joint model is used to re-rank candidate mentions and entity links produced by base recognition and linking models. Luo et al. [147] also propose a method that takes into account the mutual dependency between entity recognition and entity linking. If their entity recognition component is highly-confident on its output of entity boundaries and types, it will encourage the linking of an entity which is consistent with this output. The approach introduced in Mohapatra et al. [161] that jointly addresses linking and typing and that we have discussed earlier in Section 2.1.2 also belongs to this category.

Relation to other tasks

Entity linking might employ an *entity recognition* system as a way to perform mention detection. Entity linking techniques are important in enabling other tasks. Performing entity linking can help improve *document retrieval* (see Section 2.2.2). In principle, other tasks that rely on entity-document features can be improved by having a reliable entity linking system, as it would reduce the noisy features obtained by incorrect entity-document associations. Entity linking is also important to resolve entities for *relation extraction* to comlete a knowledge graph (Section 2.1.4). As we have discussed earlier, linking confidence is sometimes used as a strategy for entity discovery (Section 2.1.2)

Outlook

Interesting future directions for entity linking include the following: *linking on queries*, and *linking with sparse knowledge graphs*.

Entity linking in queries is still a challenging problem, because of the limited context available. Pantel and Fuxman [169] first consider entity linking in queries, estimating the relevance between a query string and an entity from query-click graphs. An improved approach to linking entities in queries using contextual information and semantic matching is presented in [24]. Blanco et al. [24] learn entity representation using contextual information from Wikipedia. Their approach can be extended by incorporating more contextual information from news, related queries, and trends. Learning useful signals from this contextual information is an important direction.

Entity relatedness information is an important signal for entity linking. Unfortunately, this relatedness information will be very sparse for long-tail entities. Developing alternative ways to infer entity relatedness from various sources, and integrating them for linking purposes, is an interesting and important challenge. Additional sources than can be used to complement such sparse knowledge graphs can be obtained by retrieving documents from the Web.

2.2.2 Document retrieval

There is little work in leveraging knowledge graphs to improve document retrieval directly. Understanding how to leverage entity annotations of text to improve ad hoc document retrieval is an emerging and open research area. Here we discuss some of the existing attempts of leveraging knowledge graphs for retrieval.

In this section, we discuss work that leverages entity-oriented information to improve document retrieval. We formally define document retrieval in this setting as follows.

Definition 16 (Document retrieval) *Given a query q and a collection D, retrieve and rank documents m that are relevant to the query.*

Approaches

Approaches to document retrieval that leverage entity-oriented information can be grouped as follows: *expansion-based*, *latent factor modeling*, and *language modeling approaches*.

Expansion-based approaches Some of this work can be considered as a variant of query expansion. Dalton et al. [55] propose to employ query expansion techniques called entity query feature expansion (EQFE) which enriches the query with features from entities and their links to knowledge bases, including structured attributes and text. They experiment with both explicit query annotations and latent entities.

They introduce EQFE, *entity query feature expansion* model, which works as follows:

- **Preprocessing** First, documents are preprocessed with entity linking and additional information obtained from knowledge graphs are indexed as different fields of the document.

- **Query annotation** At query time, the query will also be preprocessed with entity linking, providing annotations for all entity mentions in the query.

- **Expansion from feedbacks** Two kinds of feedback are then considered for expansion: 1) KB feedback, in which the query is issued against knowledge graph entries index to retrieve related entity distributions, and 2) corpus feedback, in which related entities are obtained from the retrieved documents.

The different expansion strategies include related words, entities, mentions, types, categories and neighbors. Each different expansion strategy can be incorporated as a field or a representation of the document. Feature weights are learned for each of these different expansions with a log-linear learning to rank approach.

To evaluate the effectiveness of their expansion method, Dalton et al. [55] consider three test collections: TREC Robust04, ClueWeb09B, and ClueWeb12B. They compare EQFE against a Sequential Dependence Model (SDM), SDM with collection relevance, and a relevance feedback model. EQFE achieves the best performance in terms of MAP on Robust04; it also obtains the best performance in terms of NDCG@20, ERR@20, and MAP on the ClueWeb12B collection. They do not obtain any improvement on CluebWeb09B.

Xiong and Callan [244] propose a simple method to improve document retrieval by using knowledge graphs for query expansion. A supervised model combines information derived from object descriptions and categories in knowledge graphs to select terms that are effective for query expansion. Their supervised method consider three features: the pseudo-relevance feedback score in retrieved objects, the pseudo-relevance feedback score in top retrieved documents annotations, and the negative divergence score between category distributions. This method consists of two main steps: object linking and term selection. In object linking, ranked lists of KG objects are generated. Two approaches were considered for object linking: issuing a query to the Google Search API, and selecting objects from FACC1 annotations in top retrieved documents.

Latent-factor approach Although also based on expansion using external data in knowledge graphs, another approach that considers entity (object) relationship as a latent space is proposed in [243]. Specifically, Xiong and Callan [243] propose a new technique for improving document ranking using external semi-structured data such as controlled vocabularies and knowledge bases. Their algorithm, EsdRank, treats vocabularies, terms, and entities from external data as objects connecting a query and documents. Evidence that is used to links query to objects and to rank documents are incorporated as features between query-object and object-document. A latent listwise learning to rank algorithm, Latent-ListMLE, learns how to handle all evidence in a unified procedure from document relevance judgments.

One key component of the method is Latent-ListMLE, a latent listwise learning to rank model. Latent-ListMLE aims to rerank an initial set of documents D with the help of related objects O and feature vectors U derived from the relationships between object and document, and feature vector V, representing the relationship between the object and the query.

Three strategies were considered to find related objects O given query q and document d: query annotation, object search, and document annotation. Query annotation selects objects that directly appear in the query, which can be done with entity linking. Object search selects objects that are textually similar to the query. Document annotation selects objects that appear in the retrieved documents D. Xiong and Callan [243] use a feature representation that is inspired by [55]. The relationships between query, documents, and objects are represented by a set of features. Features between query and objects considered include object selection score, textual similarity score, ontology overlap, object frequency, etc. Features between objects and documents include textual similarity, ontology overlap, graph connection, and document quality. The best combination of query representation and document ranking is then learned from these features.

EsdRank outperforms EQFE [55] on ClueWeb09B and ClueWeb12B datasets on almost all metrics. In addition, they learn that finding relevant objects for query and documents is very important; the results suggest that query annotation is the most effective object selection strategy in the current setting.

Language-modeling approach Raviv et al. [183] devise an entity-based language model. One particular feature of their model is that it takes into account the uncertainty inherent in the entity annotation process and the balance between using entity-based and term-based information. They apply entity linking on the text to obtain entities along with the linking confidence score estimated by an entity linking method. Based on the output of this annotation, an unigram entity-based language models over a token space can be defined. The token space includes the set of all terms in the document collection D and the set of entities which were linked at least once within a document in D.

The notion of pseudo-length of a text d is introduced as:

$$pl(d) = \sum_{t \in T} pc(t, d), \tag{2.19}$$

where $pc(t, x)$ is the pseudo-count (explained below) of term or entity token t. With this pseudo-length, the maximum likelihood estimate of token $t \in \mathcal{T}$ is defined as:

$$\theta_d^{MLE}(t) = \frac{pc(t, d)}{pl(x)}. \tag{2.20}$$

The important component in this model is the estimation of pseudo-count. Two strategies are considered: hard confidence thresholding and soft confidence thresholding. In hard thresholding, a threshold is first fixed on the confidence score of the annotation and only the linking within the above confidence threshold is considered for pseudo-counts. In soft thresholding, the confidence score of linking a particular mention is taken as count during the estimation of the pseudo count, interpolated with a term versus entity token importance parameter:

$$pc(t,d) = \begin{cases} \lambda c_{term}(t,d), & \text{if } t \in \mathcal{V} \\ (1-\lambda)\sum_{m \in M} \rho(m) & \text{if } t \in \mathcal{E}, \end{cases} \tag{2.21}$$

where $c_{term}(t,d)$ is the count of a term t within the document d and $\rho(m)$ is the linking confidence score of a mention m if t is an entity. Documents are the ranked by the cross entropy between the language models induced from the query q and d.

Raviv et al. [183] perform retrieval experiment on the AP, Robust, WT10G, GOV2, and ClueWeb Category B corpora. The experimental results indicate that the entity-based language model with hard and soft thresholding improves over the standard term-based language model. They also learn that their methods are effective with different entity linkers.

Relation to other tasks

Document retrieval is closely related to *entity linking* (Section 2.2.1) , as approaches to document retrieval that use entity information primarily depend on performing entity linking on the queries and the candidate documents. In another direction, document retrieval might be useful for some *entity retrieval* (Section 2.2.3) or entity recommendation (Section 2.2.4) methods, in which relevant documents are first retrieved, and then entities in these documents are ranked.

Outlook

With the emergence of deep learning, we expect more *neural entity-enhanced document retrieval* methods to emerge. One general strategy would be learning the representation of documents, queries, and entities, and using the representations to improve document retrieval in combination with more traditional term-based methods. Another interesting direction is *extracting the relationships between documents as a graph* and learn the embedding of entities from a graph to improve document retrieval [249].

2.2.3 Entity retrieval

Originally introduced through the expert finding task [12], entity retrieval has evolved into different incarnations and settings: ranking entities as found in the document collections, on knowledge graphs, or both. A TREC track has been devoted to entity retrieval [13]. We formally define entity retrieval as follows:

Definition 17 (Entity retrieval) *Given a collection D, retrieve and rank entities e with respect to their relevance to a query q.*

Forms of entity retrieval include the following: *term-based entity retrieval*, *ad-hoc object retrieval*, and *list retrieval*. We focus on the first two forms here, and consider list retrieval as form of recommendation, to be discussed in the next section.

Approaches

We grouped approaches to entity retrieval as follows: *language modeling*, *neural language modeling*, and *multi-fielded document* approaches.

Language modeling approaches Language modeling approaches span from work on expert finding [12]. They introduce two models for ranking entities given a query with two strategies: representing an entity as a virtual document, and ranking the documents mentioning the entities.

Neural approaches In the setting of expert finding, Van Gysel et al. [231] introduce an unsupervised discriminative model for the expert finding task. They learn distributed word representations in an unsupervised way, constructing it solely from textual evidence. They learn a log-linear model of probabilities of a candidate entity given the word. Later on, Van Gysel et al. [230] improve their approach to learn term and entity representations in a different space, but adjusting the representation so that they are close in the entity space.

Van Gysel et al. [230] confirm the effectiveness of their approach (Latent Semantic Entities - LSE) for retrieval when used in combination with query independent features and the query likelihood model. LSE outperforms other latent vector space baselines (i.e., LSI, LDA, and word2vec) for lower-dimensional vector spaces. One key insight from their work is that this neural approach and term-based retrieval make very different errors. In some cases, the retrieval performance is really improved by the semantic matching capability provided by LSE.

Multi-fielded document approaches One popular method for entity retrieval is using *fielded representation*: an entity is represented as a set of fields with bag-of-words values. Proposed methods then try to learn the appropriate weight for each field. The general framework for entity retrieval in the presence of structured and semi-structured data is the following: represent an entity as a fielded-document, and learn a ranking function that uses the relevance from each of the fields to obtain a final ranking [178]. Each approach discussed below tends to have its own representation and retrieval strategies. This approach is introduced in the context of ad-hoc object retrieval.

A form of entity retrieval is the *ad-hoc object retrieval task*, which utilizes link information from the knowledge graph. The goal is to retrieve a list of resource objects (i.e., entities) with respect to a user query. Pound et al. [178] defines the formalization, setting, and experimental setup for this task. One simple baseline for this task in a graph-based setting is simply TF-IDF over the entity properties in the graph, e.g., IDF statistics are computed for each properties of the entity in the graph (here the graph can be represented as an RDF graph).

Tonon et al. [224] propose a hybrid approach that combines IR and structured search techniques. They propose an architecture that exploits an inverted index to answer keyword queries and a semi-structured database to improve search effectiveness over a linked data graph. Each object in the graph is represented with the following pieces of information: entity name in URIs, entity name in labels, and attribute values of the entity. This information is later indexed as a structured, multi-fielded index of which later multi-fielded retrieval algorithms such as BM25F can be employed. The additional benefit of having a graph structure is that additional relationship data can be used as a context to improve object retrieval. Tonon et al. [224] incorporate additional method based on query expansion and relevance feedback on the graph data, and apply this in combination with the basic BM25F ranking. Experiments carried by Tonon et al. [224] indicate that the use of structured search on top of standard IR approaches can lead to significantly better results. Graph-based extensions of a baseline ranking obtain up to a 25% improvement of MAP over the baseline.

Addressing a similar task, Zhiltsov and Agichtein [261] integrate information from latent semantics to improve entity search over RDF graphs. They combine the compact representation of semantic similarity and explicit entity information. They represent an entity with common fields such as: names, attributes, and outgoing links. In addition to these common fields, the entity graph is represented as a tensor. They factorize the tensor into a number of latent factors, and enrich the current fielded representation of the entity with top related entities obtained through the latent factor modeling.

Since term dependencies models have been known to be more effective than unigram bag-of-word models for ad-hoc document retrieval, Zhiltsov et al. [262] attempt to adapt the idea for entity retrieval. They propose FSDM, a term dependence retrieval that performs ranking as follows:

$$P(D|Q) = \lambda_T \sum_{q \in Q} \hat(f)_T(q_i, D) + \lambda_O \sum_{q \in Q} \hat{f}_O(q_i, q_{i+1}, D) + \lambda_U \sum_{q \in Q} \hat{f}_U(q_i, q_{i+1}, D), \tag{2.22}$$

where λ is the mixture parameter, $\hat{f}_T, \hat{f}_O, \hat{f}_U$ are the potential functions for the unigram, ordered bigrams, and unordered bigrams, respectively. The mixture parameters and the field weight parameters are learned with the Coordinate Ascent algorithm. Later on, Nikolaev et al. [167] extended the model from [262] by generalizing it further: instead of learning the field weight parameters directly, the dependencies between the query term and field are taken into account and parameterized as a set of features based on the contribution of query concept matched in a field towards the retrieval score. The features that are used are collection statistics, part-of-speech features, and proper noun features.

Nikolaev et al. [167] propose PFSDM and PFFDM, the parameterized version of FSDM [262]. Experimental results indicate that the parameterization help improves the performance over FSDM. Taking into account both term dependencies and feature-based matching of query concepts to fields are beneficial. Parameterizing the field importance weight results in more improved topics and greater magnitude of improvements.

Hasibi et al. [95] aim to exploit entity linking for retrieval. They introduce ELR, a component that can be applied on top of any term-based entity retrieval model based on the Markov Random Field framework. They extend the Markov Random Field approach and incorporate entity annotation into the retrieval model, similar to the FSDM model introduced in [262] with an additional term that weights the importance of entity annotations; this introduces entity-based matching in addition to the term-based matching.

Hasibi et al. [95] evaluate the effectiveness of their approach on the DBPedia entity collection [10]. They compare their approach to state-of-the-art object retrieval methods such as SDM and FSDM [262]. Experimental results confirm the effectiveness of ELR when applied on top of these retrieval methods. The improvement obtained is between 6.3-7.4% in terms of MAP and 4.5-6.1% for P@10. Their results also indicate that ELR especially benefits complex and heterogeneous queries.

Graus et al. [88] propose a method for enhancing the representation of an entity from various external sources, which allows the adjustment of the entity representations to a defining characteristic of an entity at a given time. Their method allows the adjustment of each field's importance in an online manner, learning them from user click feedbacks. Graus et al. [88] consider the following static and dynamic description sources to build dynamic representations of entities:

- **Knowledge base**: four type of information are considered to build a representation from a KB, i.e., anchor text, redirects, category titles, and titles of linked entities,
- **Web anchors**: anchor text of links to Wikipedia pages,
- **Twitter**: tweets with links to Wikipedia pages,
- **Delicious**: references to entities through social tags,
- **Queries**: queries that can be linked to Wikipedia pages.

Entities are modeled as fielded documents:

$$e = \left(f^e_{title}, f^e_{text}, f^e_{anchors}, \ldots, f^e_{query}\right), \tag{2.23}$$

where f^e is the term vector that represents e's content from an description source. One particular feature of their approach is that the fields are updated over time. At every time point, the term vectors are updated with resources obtained from queries, tweets, and tags. Based on this dynamic representation, feature weights are learned for query-field similarity. field importance, and entity importance score based on each field and description.

Graus et al. [88] demonstrate that incorporating dynamic description sources into a collective entity representation allows a better matching of users' queries. They also show how continuously updating the ranker leads to improved ranking effectiveness.

Relation to other tasks

Entity retrieval depends on having a reliable *entity recognition* (Section 2.1.1) and/or *entity linking* systems (Section 2.2.1). There are similarities between entity retrieval and *entity recommendation*, which we will discuss in the next section. Recently, entity retrieval has been used as a query understanding strategy to support *document retrieval* (see Section 2.2.2).

Outlook

Future research directions on entity retrieval include the following. First, it would be interesting to combine the term-based collective representation introduced in [88] with neural representation methods. Secondly, entity representations can also be enriched by incorporating the output of document filtering systems (see Section 2.1.3).

2.2.4 Entity recommendation

Another entity-oriented task deals with recommending related entities in response to a query. We refer to this line of work as *entity recommendation*. In the literature, this task is also sometimes referred as *related entity suggestion* or *related entity ranking*.

The origin of this task can be traced back to work on ranking related entities, introduced in TREC [13]. The recommendation at this stage is typically accommodated with a description of the expected related entities as a query. Later on, more approaches are developed for the Web domain, in which recommendations are used for enhancing the results of the search engine result page. In this setting, the input is likely only an entity or an entity plus a context word.

We define the entity recommendation task more formally as follows:

Definition 18 (Entity recommendation) *Given a collection D and/or entity relations R, rank entities e based on their relatedness to a query q.*

Depending on the setting, the query q can be in the form of an entity e_q or entity e_q plus keywords/description c. Initial models are based on the entity only, without considering the context of the occurrence.

Approaches

There are three general approaches to entity recommendation: *heuristic*, *behavioral*, and *graph-based* approaches. Since the approaches that we will discuss are designed for different domains and settings, they are not directly comparable. We briefly discuss the performance of some of the approaches.

Heuristic approaches Early work on ranking related entities is based on simple statistical associations. Bron et al. [36] introduce a related entity ranking method utilizing simple co-occurrences. They first apply and compare difference co-occurrence statistics, such as Maximum Likelihood Estimation (MLE), Pointwise Mutual Information (PMI), and Log Likelihood Ratio (LLR). As an example, the MLE model is estimated as follows:

$$P(e|E) = \frac{cooc(e, E)}{\sum_{e'} cooc(e', E)}, \tag{2.24}$$

where $cooc(e, E)$ is the count of the co-occurrence of e and E in a document collection.

Later on, they incorporate a type filtering and a context model based on the terms surrounding the entities, ranking the entities as follows:

$$P(e|E, T, R) \propto P(R|E, e) \cdot P(e|E) \cdot P(T|E), \tag{2.25}$$

where $P(E|E, e)$ is the context model learned from language model of co-occurring entities, and $P(T|E)$ a type filtering model estimated from entity category assignments in Wikipedia. When evaluated within the Related Entity Finding (REF) framework [13], type filtering and context model are shown to be effective. They improve the performance of co-occurrence models by up to 115% in terms of R-precision and 29% in terms of Recall@100.

Behavioral approaches This group of approaches utilizes user feedbacks in the form of clicks on related entities, documents, or entity panes in combination with other features for recommendation. Kang et al. [119] propose a machine-learned entity ranking model that leverages knowledge graphs and user data as signals to facilitate semantic search using entities. The approach jointly learns the relevance among the entities from click data and editorially assigned relevance grades. They use click models to generate training data to learn a pairwise preferences of *entity facets*, i.e., a collection of related entities belonging to the same group. Once the facets are ranked, the related entities within each facets are ranked with feature-based models.

Continuing on this line of work, Blanco et al. [23] propose a learning to rank framework for entity recommendation based on various signals. They extract information from data sources such as Web search logs, Twitter, and Flickr. They combine these signals with a machine learned ranking model to produce a final recommendation of entities to user queries. They use features based on co-occurrence, entity popularity, knowledge graph, and other features such as entity types and entity relations. They model the task in a learning to rank framework, learning a function $h(\cdot)$ that generates a score for an input query q_i and an entity e_j that belongs to the set of candidate entities related to the query E_q. They represent q_i and e_j as feature vector w.

Blanco et al. [23] evaluate the recommendation performance by collecting judgments on the related entities output. They achieve an NDCG@5 score between 0.824-0.950 accross different entity types. In addition, they find that the type of the relation is the most important one for entity recommendations.

The next set of approaches considers the user profile, essentially estimating $p(e|u)$, while other models attempt to rank entity based on context words, estimating $p(e|q)$, or a more complex combination of these two tasks. In contrast to the previous recommendation model, which recommends entities given an entity query, a contextual model now incorporates information such as the profile, or text currently selected/browsed by the user.

Yu et al. [255] utilize information from user click logs and knowledge extracted from Freebase. They propose some heuristics and features for the task and propose a generic, robust, and time-aware personalized recommendation framework to utilize the heuristics and features at different granularity level. Their method utilizes various pairwise similarity measures extracted from both user log dataset and knowledge graph. Their method considers the consistency and the drifting nature of user interests, different types of entity relationships as well as several other heuristics. They propose to include knowledge graph features such as path features, relationship features, content similarity features, and co-clicks. Most of these features are pairwise features derived from the main entity and the candidate related entity. They incorporate pointwise features such as co-click, global popularity, current popularity, and cross-domain correlation.

Later on, Yu et al. [256] consider personalization when generating their entity recommendation. They propose a graph-based approach, using the heterogeneous information network to link entities and users to generate personalized recommendations. They learn a recommendation model for each user based on the users' implicit feedback. To handle sparsity, they first discover the groups of users which have similar preferences and used these groupings to learn an aggregated, personalized recommendation model. The final recommendation is generated by a combination of the user-based and group-based model

Related to the previous method that uses knowledge graphs and click logs data, Bi et al. [20] include a novel signal: *entity pane* log. They propose a probabilistic entity model that provides a personalized recommendation of related entities using three data sources: *knowledge base*, *search click*, and *entity pane log*. Specifically, their model is able to extract hidden structures and capturing underlying correlations among users, main entities, and related entities. Furthermore, they incorporate clickthrough signal for popular entities, extracting three types of clickthrough rates (CTRs): CTRs on related entities, CTRs on main entities and related entities, and CTRs on users, main entities, and related entities. They use the feedback from the entity pane to estimate the likelihood of the data and generate training labels. The observation is a set of triples (u, m, r) which represents clicks from user u on a related entity r and a main entity m. They learn the preference between triple pairs to estimate the parameters of their models. More specifically, training data is created by assigning positive class labels to clicked triples, and negative class labels to non-clicked triples. Instead of learning the labels directly, the learning is to learn preference between the clicked triples and non-clicked triples.

The following method utilizes behavioral signals within a deep learning framework. Gao et al. [82] propose a method that observes, identifies, and detects naturally occurring signals of interestingness in click transitions between source and target documents, collected from commercial Web browser logs. After identifying the keywords that represent the entities of interest to the user, they aim to recommend other, interesting related entities. They model interestingness as learning a mapping function that quantifies the degree of interest that a user has after reading a source document. They train a deep semantic similarity model on Web transitions and map source-target document pairs to features vectors in a latent space such that the distance between the source document and the corresponding target in that space is minimized.

Graph-based approaches This group of approaches relies primarily on the connections of entities in a graph to generate recommendations. Bordino et al. [32] explore the entity recommendation problem by focusing on the serendipity aspect of recommendations. They set to seek the answer to what makes a result serendipitous by exploring the potential of entities extracted from two sources of user-generated content: Wikipedia an Yahoo! Answers. The context of each data source is represented as an entity network, which is further enriched with metadata about sentiment, writing quality, and topical category. They extract entity networks from each dataset by the following procedures. First, they extract a set of entities. They construct an entity network by using a content-based similarity measure to create links between entities.

To generate recommendations, they perform a random walk with restart to the input entity. They take an input graph, a self-loop probability β, and a start vector defined on the nodes of the graph which in this case contains only the input entity. The random walk starts in the node corresponding to such entity.

Also using the random walk framework, Bordino et al. [31] consider the task of entity-oriented query recommendation given a web page that a user is currently visiting. First, they represent the topics of a page by the set of Wikipedia entities mentioned in it. To obtain query recommendation, they propose a novel graph model: the entity-query graph, which contains the entities, queries, and transitions between entities, queries, and from entities to queries. They perform personalized PageRank computation on such a graph to expand the set of entities extracted from a page into a richer set of entities, and to associate these entities with relevant query recommendations.

Lee et al. [127] present a contextual entity recommendation approach for retrieving contextually relevant entities leveraging knowledge graphs. Contexts such as user text selection and the document currently browsed by the user are incorporated for recommendation. An undirected graph of entities where there is an edge (x, y) if there is a link to an entity y on the Wikipedia page of the entity x or vice versa. For recommendation, they create a subgraph containing the user-selected entity, entities in the document, and a set of candidate entities. Finally, they rank candidate entities by combining their betweenness and Personalized PageRank scores.

Similar to the previous work, Fuxman [80] deals with entity recommendation given that a user is currently reading a particular article and select a portion of the text in an article. The problem also considers the quality of selection made by the user, and whether this selection is intentional or accidental. He first identifies a set of candidate references. Then, he learns a prediction function $p(d|s, c, D(s, c)$ to score each candidate given a text selection s, the full content of the document c, and the content of the candidate document d. Lastly, they recommend a candidate concept if the score is above a threshold δ. For learning, he utilizes the MART algorithm with 27 features derived from three criteria: *context coherence*, *selection clarity*, and *reference relevance*.

Relation to other tasks

Entity recommendations have a strong connection to *entity retrieval* (Section 2.2.3). The main distinction is that in retrieval we are retrieving entities relevant to a query, i.e., to answer the query; in recommendation we recommend entities related to the query.

In addition, *entity linking* (see Section 2.2.1) also forms the building block of entity recommendation. For example, in [168] the task of related content finding can be considered as a form of recommendation. Specifically, the task is finding video content related to a live television broadcast, leveraging the textual stream of subtitles associated with the broadcast. The query for recommendation is obtained by linking entities in the subtitles of the video.

Outlook

Interesting research directions for entity recommendation include the following: *encouraging explorative behavior*, *leveraging heterogeneous information*, and *context-specific entity recommendation*.

In a follow-up work to [23], Miliaraki and Blanco [157] conduct an in-depth analysis on how users interact with the entity recommendation system. They characterize the users, queries, and sessions that appear to promote explorative behavior. Taking this idea one step further would be to develop entity recommendation systems that enhance serendipitious once such explorative behavior is detected.

Zhang et al. [259] propose an approach to leverage the heterogeneous information in a knowledge graph to improve the quality of recommender systems with neural methods. They adopt TransR (see Section 2.1.4) to extract items' structural representations by considering the heterogeneity of both the entities and relationships. Besides this representation learning method, heterogeneous information encoded as graphs can also be leveraged by designing recommendation algorithms that rely solely on the semantic of the connections.

Beside purely exploratory purposes, users might have a specific recommendation goal or context when using entity recommendation systems. Formalizing these different goals and translating them into objective functions that can be optimized in the context of recommendation is an interesting challenge.

2.2.5 Entity relationship explanation

Entity relationship explanation is also a new, emerging task. Fang et al. [70] first introduce the task of entity relationship explanation. The motivation was that, observing some entity relations from user data such as query logs, some common entity pairs arise. Explanations are required to describe the connections between these two entities.

We formalize the task of entity relationship explanation as follows:

Definition 19 (Entity relationship explanation) *Given a pair of entities e and e', explain why they are related.*

Approaches

Two approaches for generating explanation exist: *instance-based explanations* and *description ranking*.

Instance-based explanation Fang et al. [70] focus on explaining the connections utilizing knowledge graphs, specifically. They mine relationship explanation pattern, which is modeled as a graph structure, and generate an explanation instance from this pattern. Their approach consists of two main components: explanation enumeration and explanation ranking.

They rely on path enumeration in the explanation enumeration phase, generating all path instances of a specified length. Two path instances will be connected if they end at the same node. This enumeration step will result in a number of paths, which will be combined to form a minimal explanation. They propose two kind of interestingness measures that can be computed from the candidate paths: structure-based measures and aggregate measures. Structure-based measures are obtained from the topological structure of the explanation pattern, e.g., the size of the pattern. Aggregate measures are obtained by aggregating over individual explanation instances. This includes statistics such as counts of explanation instances. The aggregate measures are then normalized to obtain distribution-based measures. They idea is to estimate the rarity of an explanation. Finally, the explanation candidates will be ranked by any of the previous individual measures.

Seufert et al. [204] propose a similar approach on entity sets, although working towards a slightly different task. Their method focuses on explaining the connection between entity sets based on the concept of relatedness cores: dense subgraphs that have strong relations with both entity query sets. This dense subgraph is expected to represent key events in which the entities in the sets. It is aimed to find multiple substructures in the knowledge graphs that are highly informative. Their approach relies on two phases: finding relationship centers and expanding the relationship centers into relatedness core. Relationship centers are intermediate vertices that play an important role in the relationship. These relationship centers must be connected to both query sets. They are identified by performing random walks over the graph, adapted from the Center Piece Subgraph method [223]. Once the relationship centers are identified, the subgraph will be expanded to obtain the relationship core. The relatedness core is built in three steps: obtaining entities related to the center, expanding these subgraphs with entities related the entities in the subgraph, and finally adding entities that are related to the query entities.

Explanations in the form of graph output are then assessed by human annotators. Pairs of subgraph are presented to the annotators, then the annotators are tasked to give their preferences. Seufert et al. [204] is shown to be better than the baseline Center Piece Subgraph [223] and also [70].

Description ranking Voskarides et al. [234] study the problem of explaining relationships between pairs of knowledge graph entities, but aim to do so with human-readable descriptions. They extract and enrich sentences that refer to an entity pair, then rank the sentences according to how well they describe the relationship between the entities. They model the task as learning to rank problem for sentences and employ a rich set of features, instead of individual interestingness measures as proposed in [70].

The approach introduced in Voskarides et al. [234] requires a document collection containing the entities. They split either entities' Wikipedia article Wikipedia articles into sentences and extract sentences as candidates if they contain the surface form of the other entities, or sentences containing both entities' surface forms or links. To make candidate sentences readable outside the article, they perform sentence enrichment by performing pronoun resolution and linking.

Candidate explanation sentences will be ranked by how well they describe a relationship of interest r between entities e_i and e_j. To combine various signals, each sentence is represented as features and a learning to rank approach is employed. The features considered are the following: text, entity, relationship, and source features. A Random Forest classifier is then used to learn a ranking model.

For evaluation, candidate sentences are judged in four relevance grades. Ranking-based metrics such as NDCG and ERR are then computed on the description ranking using these judgments. The best variant of the model introduced in Voskarides et al. [234] achieves an NDCG@10 score of 0.780 and an ERR@10 score of 0.378.

Relation to other tasks

Relation explanation is important in the context of *entity recommendation* (see Section 2.2.4), as they will allow users to asses the output of entity recommendations models better. As for dependencies, relation explanation methods which rank external text descriptions rely on having entity recognition and classification (Section 2.1.1), and/or entity linking (Section 2.2.1) performed on the text.

Outlook

As to future directions, we expect more *complex explanation models* (e.g., neural models) to emerge, able to provide explanations for *directly and indirectly connected entities*; and also explanation for *a group of related entities*. More approaches that rely on *text generation* instead of existing description are also likely to emerge.

3
Entity Network Extraction

In this chapter, we start the exploration of the first theme of this thesis: *entity-entity associations*. The setting is as follows: we have a document collection and need to extract associations from documents contained in the collection. We consider a scenario where there exist a query entity and a set of its related entities, for which the associations between the pairs of entities might be of different strengths. At this point, we ignore the nature (i.e., type) of the relationships between pairs of entities, and estimate the strength of each relation by aggregating its individual occurrence in the document collection.

The method introduced in this chapter is aimed towards supporting researchers in the humanities and social sciences domain by facilitating new exploratory search options based on entity-entity associations. Today's increasing digitization and curation of humanities content in digital libraries gives rise to a new and interesting set of opportunities. In computational humanities, researchers are particularly interested in applying computational methods and algorithms to gain insight from this kind of data [146]. One interesting and urgent problem is extracting and analyzing networks of entities from unstructured, possibly noisy text such as (archival) newspaper articles. Recognizing such entities (person, organization, or location) and discovering how they are connected to each other benefits computational humanities researchers asking questions about network and entities, for example in understanding the network of an elite politician and its dynamics [71].

We view entity network extraction task as a form of semantic search. Our working hypothesis is that having entities and related entities presented in the form of a network is more useful than returning a large list of documents and forcing users to go through each and every one of them to manually identify the connections. For our purposes a network is a graph with a main entity together with a set of related entities as nodes, with edges connecting these nodes. A connection between two nodes denotes that there is a relationship between these two entities according to evidence found in the text. In our computational humanities application scenario, our users use a manually constructed English corpus of newspaper articles about Indonesia collected over a 10 year period. This amounts to 140,263 articles, mostly consisting of politics and economy articles. Figure 3.1 shows (part of) an entity network automatically extracted from the corpus. The query entity is "BJ Habibie," a former president of Indonesia. Because the query entity is a popular person, he is related to many other entities in the text. We rank the entity relations based on a scoring method, and build the network from top ranked entities only. Although we use an English-language corpus with Indonesian politics as the primary topic, the

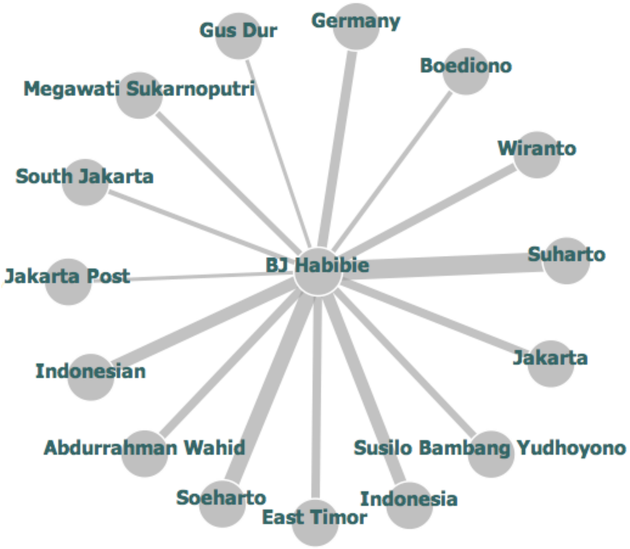


Figure 3.1: A sample network retrieved in response to “BJ Habibie” as query entity. The thickness of the links depicts the association strength as represented in the document collection.

approach proposed in this chapter also works for other languages with minor changes in the pipeline. Our approach does not rely on domain-specific pattern extraction, so it will be adaptable to other topics or domains as well.

The closest benchmarking task to our proposed task is the related entity finding (REF) task that was considered at TREC 2009, 2010 and 2011 [14]. Related entity finding works as follows: given a source entity, a target page, a narration of the relation of interest, one has to give a ranked list of entities and their home pages that engage in this relation with the source. The task that we propose in this chapter is different from the related entity finding task in the sense that we only have the names of the entities; no sample homepage, and no narration. Furthermore, we are not interested in a single specific relation, but in all possible relationships. We ask the following question:

RQ1 How do we rank related entities to support the exploration of a document collection relying on signals from the text alone?

In this chapter we address the task of extracting an entity network from text in two ways: (1) by discovering associations between entities through statistical or information-theoretic measures, and (2) by performing relation extraction and building a network using the relationships discovered. We contrast these two approaches and also consider a combination of the two types of approach based on pairwise learning-to-rank [116].

The remainder of this chapter is organized as follows. We discuss related work in Section 3.1. In Section 3.2, we describe our proposed method. The experimental setup is detailed in Section 3.3. We follow with results in Section 3.4 and conclude in Section 3.5.

3.1 Related work

Entity Network Extraction as Semantic Search. Previous research has dealt with extracting various kinds of network from document collections. Referral Web [120] takes a person name as input and finds people related to this person on the Web by using an external search engine. Referral Web uses the number of pages where two person names co-occur to measure the degree to which they are related.

Merhav et al. [154] perform extraction of relational networks of entities from blog posts. This is done by first creating entity pairs, clustering those entity pairs, and later labeling these clusters with the nature of the relationship. Elson et al. [66] extract social networks from literary fiction. The networks are derived from dialogue interactions, thus the method depends on the ability to determine whether two characters are in a conversation. Their approach involves name chunking, quoted speech attribution, and conversation detection. Tang et al. [219] extract social networks of academic researchers. After entities are identified and disambiguated, they provide a shortest-path search mechanism that links the researchers and their publications as a network.

Association Measures Association measures can be used to describe the relationship between two words or concepts. There are various ways to measure associations or relatedness. We distinguish between the following types: frequency-based, distance-based, distributional similarity/feature-based, and knowledge-based measures.

Frequency-based measures rely on the frequency of word co-occurrences and the (unigram) frequency of each word. These include measures that are derived from probability theory or information theory, for example Chi-Square, Pointwise Mutual Information, and Log Likelihood Ratio [48]. Distance-based measures rely on the distance between words in the text. Co-dispersion, introduced in [239], is one such measure.

Feature-based or distributional similarity measures describe the relatedness between two words or concepts based on the distribution of words around them. These are measures based on extracting a number of features for each entity, and then comparing the feature vectors for different entities. One example is by using cosine similarity to determine the relatedness of two entities based on linguistic features, such as neighboring words, part-of-speech tag, etc. [48]. Knowledge-based measures are measures that use an ontology, thesaurus, or semantic network to determine the relatedness between words or concepts [158].

Relation Extraction In relation extraction, we want to extract relations between entities such as persons, organizations, and locations. *Supervised* methods view the relation extraction task as a classification task. Features are extracted from entity pairs and a classifier is trained to determine whether a pair of entities is related. There are various groups of methods: feature based methods, in which syntactic and semantic features are extracted from the text, and string kernel methods, where the whole string is passed as a feature and string kernel functions are used to recognize the richer representations of the structure within the strings to determine whether two entities are in a relation.

Semi-supervised methods are often based on pattern-based extraction algorithms. The core idea is bootstrapping, in which one tries to extract patterns iteratively, using newly found patterns to fuel later extraction steps. DIPRE [35] starts with a small set of entity pairs; the system then tries to find instances of those seeds. With newly found instances, the relation is generalized. Snowball [1] uses the same core idea. Snowball starts with a seed set of relations and attaches confidence scores to them; it uses inexact matching to cope with different surface structures. TextRunner [17] learns relations, classes, and entities from text in a self-supervised fashion. The system starts by generating candidate relations from sentences, then uses constraints to label candidates as positive or negative examples to feed a binary classifier.

Since labeling and annotating a corpus to create relation examples is an expensive and time-intensive procedure, there is increasing attention for *unsupervised* or *weakly-supervised* approaches to relation extraction. With distant supervision [160], indirect examples in the form of relations from a knowledge base such as Freebase and DBPedia are used. From these relation tuples, instances of relations in the form of sentences in the corpus are searched. From these sentences, text features are extracted that are then used to train classifiers that can identify relations.

Our work differs from the related work described above in the following important ways. Firstly, in building the network, we also look at measures to determine the score of the related entities. Secondly, we experiment with alternative association measures, i.e., distance-based ones. Thirdly, while relation extraction methods usually train a specific classifier for each predefined relation type, we train a generic relation classifier on linguistic features. To the best of our knowledge, we are the first to consider combining association finding and relation extraction to extract an entity network from text.

3.2 Method

Task Description The task of network extraction is as follows: given a corpus and an input entity as a query, we must return a list of related entities, along with scores that can be used to rank them. The scores can be used for visualization purposes, and can be interpreted as the strength of association between the entities, or number of pieces of evidence supporting an extracted connection.

Pipeline In the preparation stage, we enrich each document with linguistic annotations. We perform the following types of linguistic processing: tokenization, part-of-speech tagging, sentence splitting, constituency parsing, and named entity recognition with the Stanford NLP tools [122]. We later construct an index out of these documents and their linguistic annotations.

Our main pipeline consists of the following steps: (1) query construction, (2) document selection, (3) entity extraction, (4) candidate scoring, and (5) candidate ranking.

(1) Query Construction For each query entity e, we construct the query q, a phrase query that will be used in searching the index.

(2) Document Selection For retrieval purposes in the search step, we use Lucene,[1] which combines a boolean model and vector space model. After obtaining the search results, we use all of the returned documents in the next step.

(3) Entity Extraction For every document in the search result, we extract pairs of entities (x, y) that co-occur within the same sentence. We then filter these pairs of entities, to only consider pairs that contain the query entity e.

In filtering the pair of entities, we follow the rule-based inexact matching scheme used in expert finding [15], but we adapt the rules to suit our task:

- EXACT MATCH returns a match if x is mentioned exactly the same as query entity e.
- LAST NAME MATCH returns a match if x is the last name of the query entity e.
- FIRST NAME MATCH returns a match if x is the first name of the query entity e.

(4a) Candidate Scoring – Association Measure A score is assigned for each entity pair based on association measures. We compute the association strength by several *frequency based measures*: pair frequency, pointwise mutual information (PMI), and Jaccard. In the following equations, $f(x, y)$ denotes the frequency of two entities appearing together in the same sentence, $f(x)$ is the unigram frequency of entity x within the set of selected documents, and $f(y)$ is the unigram frequency of entity y within the set. Pair frequency is computed as follows: $PF(x, y) = f(x, y)$. Pointwise mutual information is computed as follows: $PMI(x, y) = \log \frac{f(x,y)}{f(x)f(y)}$. The Jaccard measure is computed as follows: $Jaccard(x, y) = \frac{f(x,y)}{f(x)+f(y)-f(x,y)}$. Both document-level and sentence-level frequency are used as evidence in counting the frequency. With document-level frequency as evidence, $f(x, y)$ is basically the document frequency of entity pairs.

We also experiment with *distance-based measures*, first by simply using the average distance of two entities. Here distance means the number of tokens separating two entities. With M denoting mean, we define the *inverse mean distance* (IMD) as follows: $IMD(x, y) = \frac{1}{M(dist_{xy1},\ldots,dist_{xyn})}$, where $dist_i$ is the linear word distance at the pair occurrence i.

An alternative to linear word distance is dependency distance. To get a dependency distance, we first need to perform dependency parsing [74] on sentences containing the entity pair. The result of this parsing is a dependency tree. Entities are not stored in a single node in a parse tree, but broken down into component words. We define dependency distance as the number of edges between the head word of entity x to the head word of entity y. We find the shortest path between these two head word nodes, and use the number of edges as distance. We then simply subsitute dependency distance as $dist$ in the previous equation to compute the dependency-based IMD.

Based on the preliminary observation that simply using pair frequency performs quite well, we propose the following measure: $PF.IMD(x, y) = PF(x, y) \times IMD(x, y)$. This measure takes into account both frequency and average distance. The intuition behind this is that a good relation will spread across a lot of documents with small dependency distance.

[1] http://lucene.apache.org

(4b) Candidate Scoring – Relation Extraction We use sentences containing the pairs of entities as text snippets. We extract the following features from each text snippet: named entity types, dependency distance, linear distance, typed dependencies (conjunction, noun modifier, or preposition), dependency trigram/bigram, and punctuation type between entities. Sentence level features are also extracted: number of tokens, the presence of quotes, and number of entities within the sentences. We avoid using lexical features in order to have a domain-independent, generic classifier.

We use a portion of our ground truth to train and tune a SVM classifier [175]. For every pair of entities that is extracted, we run the classifier to determine whether their snippets describe that the two entities are related. The snippets that are classified as correct relations will serve as $support$ instances to the relation. We score the entity pairs based on how many support instances remain after the classification. We also calculate the $confidence$ score of a pair, defined as the number of snippets detected as relations over all the snippets extracted containing the pair. We define another score as combination: $support.confidence$.

(5) Candidate Ranking We simply rank entity pairs based on the scores computed in Stage 4.

Combination Methods As we will see below, the network extraction methods that we consider behave quite differently. Because of this, we also experiment with learning to rank for combining rankings produced by various methods. Specifically, we use RankSVM [116], a pairwise learning to rank algorithm. Scores from various network extraction methods are used to build an ensemble ranking model. We try different combinations of ensembles. First, training an ensemble using scores from all methods, and also ensembles built from each family of methods. We also experiment with ensembles based on automatic feature selection. We use a filtering approach, ranking features by importance, using randomized trees [86]. Randomized regression trees are built from subsamples of the training data. Feature importance is computed based on the number of times a feature is selected as decision node in the randomized trees [175]. We use the top 4, 6, 8, and 10 features from this feature selection step to build our ensembles.

Network extraction methods compared All in all, we consider the methods listed in Table 3.1 for extracting networks.

3.3 Experimental Setup

Research Questions In the beginning of this chapter, we ask the following question:

RQ1 How do we rank related entities to support the exploration of a document collection relying on signals from the text alone?

We expand **RQ1** into the following specific questions:

RQ1.1 How do related entity ranking methods based on association measures and relation extraction compare?

Table 3.1: Entity network extraction methods considered in the chapter.

Method	Description
pf-doc	Document-level pair frequency
pmi-doc	Document-level PMI
pf-sen	Sentence-level pair frequency
pmi-sen	Sentence-level PMI
jaccard-doc	Document-level Jaccard
jaccard-sen	Sentence-level Jaccard
imd-lin	Inverse mean distance, linear
imd-dep	Inverse mean distance, dependency
pf-doc.imd-dep	Document-level PF.IMD, dependency
pf-sen.imd-dep	Sentence-level PF.IMD, dependency
rel-conf	Relation confidence
rel-support	Relation support
rel-conf.rel-support	Relation confidence.support
ensemble-all	Ensemble of all methods
ensemble-freq	Ensemble of frequency methods
ensemble-dist	Ensemble of distance methods
ensemble-freq.dist	Ensemble of frequency and distance methods
ensemble-rel	Ensemble of relation extraction methods
ensemble-top-4	Ensemble of top 4 methods from feature selection
ensemble-top-6	Ensemble of top 6 methods from feature selection
ensemble-top-8	Ensemble of top 8 methods from feature selection
ensemble-top-10	Ensemble of top 10 methods from feature selection

RQ1.2 Can we combine these various scoring methods in an ensemble to improve the performance?

RQ1.3 How does performance differ across different queries?

Dataset We use a corpus manually constructed by social historians, from web articles during the period between 2000 and 2012.[2] The corpus contains 140,263 articles about Indonesia and South East Asia. These are mainly news articles from English language media based in Indonesia such as Jakarta Post and Jakarta Globe. Some articles from international media such as The Washington Post and The New York Times are also included. The articles cover a diverse set of topics: politics, economy, cultural events, etc. Some of the named entities of the type organization and location appear in the their English version. An example of this case is “Badan Intelijen Negara” (BIN), which appears in the text both as “BIN” and “State Intelligence Agency.”

Ground Truth We prepare our ground truth by using a pooling strategy (similar to TREC [93]). We select 35 query entities that are known to occur in our corpus, run all entity network extraction methods listed in Table 3.1 and pool the top 10 related entities from each method. In the assessment step, pairs (query entity, related entity) are presented

[2] Access to the dataset and ground truth can be facilitated upon request.

to three assessors (domain experts) along with supporting text snippets. The assessors' task is to decide whether the two entities are directly related based on the text snippets containing the pair. The assessors are not given a strict definition of a relation. In case of disagreement, the majority vote determines the final assessment. We reach 80 percent average pairwise agreement between the assessors, with a kappa value of 0.60.

Evaluation Metrics and Significance Testing We use recall, precision and F-measure to evaluate the performance of our entity network extraction methods. In this task, recall is the fraction of correct relations retrieved over all relations in our ground truth. Precision is the fraction of correct relations over the retrieved relations. We mainly look at the performance in the top ten and thirty entities returned. For significance testing, we use a paired t-test with $\alpha = 0.05$.

3.4 Results

We run our entity network extraction approach on the query entities with various scoring methods. Table 3.2 shows the results of extracting the top-10 and 30 related entities.

Methods Comparison To answer **RQ1.1**, we look at the performance of the non ensemble methods. Overall, we can see that `pf-doc`, simply counting the number of documents in which the pair of entities co-occur, already provides a decent performance. Using the sentence count, `pf-sen`, further improves the performance. The `Jaccard` measures, both at the document and sentence count, perform slightly worse than `pf`. The `pmi-doc` and `pmi-sen` methods both perform significantly worse than the baseline.

PMI yields the worst performance compared to all other methods. When we look at the actual relations returned by `pmi-doc` and `pmi-sen`, we find that it is prone to extracting rare co-occurrences of entities. As a consequence, errors in the preprocessing stage (e.g., named entity recognition errors) sometimes appear in the results. Distance-based methods also perform worse than the baseline. Relying on distance alone, two entities that only appear once within close distance can easily be favored over ones that appear more often.

We take a closer look at query entity "BJ Habibie." by comparing the top-10 results of `pf-doc` and `imd-dep`. In Table 3.3 correctly related entities are shown in bold face. On this particular query, `pf-doc` clearly outperforms `imd-dep`. Almost all of the non-related entities retrieved by `imd-dep` in the table appear with the query entity in the same sentence as *enumerations* (e.g., listings of people attending a particular event). In a dependency parse tree, this type of co-occurrence will appear with dependency distance of 1, with *conjunction* as the dependency type. It is interesting to note that by using average distance instead of frequency, we successfully retrieve relations that do not occur often in the text. The two relations: "IPTN" (company founded by BJ Habibie), and "Watik Pratiknya" (a friend of BJ Habibie) are the kind of relations that are less frequently present in our corpus, since news articles are more likely to describe event-based stories instead of giving description of one's family or friends.

As we have seen, replacing frequency by distance has its own advantages and disadvantages. We proceed to look at the performance of our proposed method `pf.imd`,

Table 3.2: Results of the entity network extraction methods at top-10 and top-30 related entities. Significance is tested against the baseline with $\alpha = 0.05$.

Method	R@10	P@10	F@10
pf-doc (baseline)	0.506	0.544	0.478
pmi-doc	0.365▼	0.321▼	0.295▼
pf-sen	0.519	0.558	0.491
pmi-sen	0.328▼	0.309▼	0.281▼
jaccard-doc	0.520	0.529	0.468
jaccard-sen	0.483	0.529	0.460
imd-lin	0.434	0.355▼	0.350▼
imd-dep	0.425	0.366▼	0.347▼
pf-doc.imd-dep	0.516	0.515	0.461
pf-sen.imd-dep	0.519	0.524	0.465
rel-conf	0.365▼	0.326▼	0.312▼
rel-support	0.489	0.501	0.452
rel-conf.rel-support	0.443▼	0.429▼	0.398▼
ensemble-all	**0.569**	**0.564**	**0.507**
ensemble-freq	0.544	0.552	0.490
ensemble-dist	0.504	0.498	0.447
ensemble-rel	0.544	0.541	0.486
ensemble-freq.dist	0.470	0.475▼	0.431
ensemble-top-4	0.409	0.315▼	0.321▼
ensemble-top-6	0.439	0.349▼	0.351▼
ensemble-top-8	0.548	0.535	0.484
ensemble-top-10	0.555	0.549	0.494

Method	R@30	P@30	F@30
pf-doc (baseline)	0.775	0.324	0.435
pmi-doc	0.613▼	0.245▼	0.333▼
pf-sen	0.785	0.329	0.441
pmi-sen	0.609▼	0.241▼	0.327▼
jaccard-doc	0.763	0.318	0.427
jaccard-sen	0.763	0.323	0.431
imd-lin	0.670▼	0.257▼	0.354▼
imd-dep	0.685▼	0.268▼	0.367▼
pf-doc.imd-dep	0.803	0.334	0.449
pf-sen.imd-dep	0.815	0.342	0.459
rel-conf	0.712▼	0.277▼	0.381▼
rel-support	0.795	0.332	0.446
rel-conf.rel-support	0.777	0.321	0.433
ensemble-all	0.822▲	0.343	0.461▲
ensemble-freq	0.772	0.321	0.431
ensemble-dist	0.800	0.333	0.448
ensemble-rel	**0.825▲**	**0.346▲**	**0.465▲**
ensemble-freq.dist	0.788	0.328	0.442
ensemble-top-4	0.685▼	0.262▼	0.362▼
ensemble-top-6	0.703▼	0.271▼	0.374▼
ensemble-top-8	0.818▲	0.341	0.459
ensemble-top-10	0.820▲	0.342	0.460▲

Table 3.3: Comparing `pf-doc` and `imd-dep`.

pf-doc	imd-dep
Suharto	Taufik Kiemas
Soeharto	Wahid
Indonesian	Megawati Soekarnoputri
Indonesia	**IPTN**
Germany	Emil Salim
Abdurrahman Wahid	**Watik Pratiknya**
Wiranto	Sudi Silalahi
East Timor	Soehardjo
Jakarta	Xanana Gusmao
Susilo Bambang Yudhoyono	Sarwono Kusumaatmadja

which combines frequency and distance. This combination yields some improvement over the baseline at top-30 results, but the improvement is not significant.

With `rel-conf`, the relations that are detected by the machine learning method, but only found in one sentence, can outweigh relations that appear in many sentences. This explains why `rel-support` has a better performance, even outperforming both `pf-doc` and `pf-sen` for the top-30 results. The method `rel-support`, which can be viewed as a filtered version of `pf`, classifies text snippets before counting the frequency. This provides a more reliable way of counting the pair frequency. However, when we see the per-query results, the classifier does not always work, leading to a lower average performance compared to `pf-doc` and `pf-sen` (for the top-10 results).

Next, we contrast the results of a relation extraction method, `rel-support` with `pf-doc`, again for the query “BJ Habibie.” The relations are listed in Table 3.4. For this query, the filtering effect of the relation extraction classifier manages to improve the results. The resulting ranking introduces three new entities (all related) and pushes out one non-related entity.

Table 3.4: Comparing `pf-doc` with `rel-support`.

pf-doc	rel-support
Suharto	**Suharto**
Soeharto	Abdurrahman Wahid
Indonesian	**Indonesian**
Indonesia	**Megawati Soekarnoputri**
Germany	**Soeharto**
Abdurrahman Wahid	**Germany**
Wiranto	**Susilo Bambang Yudhoyono**
East Timor	**Boediono**
Jakarta	**ICMI**
Susilo Bambang Yudhoyono	**Golkar**

Ensemble Methods To answer **RQ1.2**, we contrast the results of our ensemble methods against the non-ensemble ones. Table 3.2 shows that most ensemble methods give improvements over the baseline. Indeed, the overall best performance is achieved using ensemble methods. The improvements are statistically significant at top 30 related entities (`ensemble-all` and `ensemble-rel`, and `ensemble-top-10`). Simply using all of the methods in one ensemble can give a good performance. Ensembles of methods within the same family do not perform as well as combining method from various families. An exception to this is `ensemble-rel`, which only combines relation extraction methods scores.

Interestingly, the tree-based feature selection returns the following as top-6 features: `imd-lin`, `jaccard-sen`, `relation-conf`, `jaccard-sen`, `pmi-sen`, and `imd-dep`. Using these top-4 and top-6 features in an ensemble results in poor performance. As we observed above, three of these scoring methods are among the worst performing methods, thus combining them without adding (many) other scoring functions reinforces the weaknesses.

Score Differences between Entities To answer **RQ1.3**, we average the performance of all methods on each query. As shown in Figure 3.2, the performance varies. Some entities appear frequently in the dataset, therefore having more possible candidates and more possible types of context and relations. However, there does not seem to be a direct correlation with entity network extraction performance.

What went wrong with the worst performing queries? The person in query-24, "J Kristiadi," is a political observer. Most sentences mentioning him in the text are statements containing his observation about other entities, while only two describe actual relations to his affiliations. On this extreme case, most methods fail. For query-26, most of the snippets consist of mentions of the query with other entities in the form of enumerations. The snippets of query-29 also contain speech statements about other entities, along with invalid snippets created due to sentence splitting errors.

As shown in Figure 3.2, query-11 has the highest average performance. The person in query-11, "Edy Harjoko," is a military commander. Most snippets in the text mention his rank or role in the organization (i.e., "TNI Chief of General Affairs Edy Harjoko"). There is almost no direct/indirect speech found in the snippets of this query. The snippets of query-23 also consist of a lot "head of" and "founder of" mentions. The next best performing query contains a lot of snippets in the form of appositions (e.g., "who founded ..."). Overall, we can say that these queries have more reliable snippets.

Error Analysis We further analyze the errors made by most methods. In particular, we look at the bottom-10 query entities for which the worst performance is observed. By inspecting the supporting text snippets, we discover several types of error, mostly caused by the type of sentence that is used to extract the co-occurrence.

One of the most common cases is sentences containing *indirect/direct speech*, in which one entity mentions other entities. The fact that one entity mentions another entity does not necessarily mean that they have a direct connection. The low performing queries tend to have more of this type of sentence than other queries, as we have shown with query-24.

Another common case of errors are *enumerations*. As we have described above,

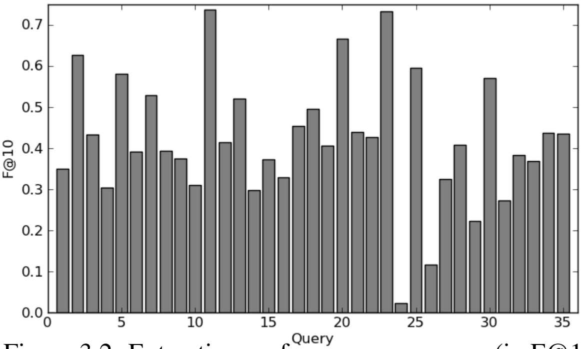


Figure 3.2: Extraction performance per query (in F@10).

enumerations of entities do not necessarily mean that the entities enumerated are related. We observe that in our document collection most enumerations are ad-hoc, i.e., listing a number of entities that attend a certain event. When the text snippets returned for a query entity contain many enumerations, we tend to get a lower performance.

3.5 Conclusion

Today, more humanities content is archived and made available in digital libraries. We have presented the task of entity network extraction from text that can be applied to these types of content. The task is studied in the context of a computational humanities application scenario. Our approach introduces an information retrieval pipeline that involves document search, entity extraction, and entity pairs scoring based on multiple scoring functions.

We have asked the following question:

RQ1 How do we rank related entities to support the exploration of a document collection relying on signals from the text alone?

To answer RQ1, we have explored various methods for retrieving and ranking entity pairs, based on co-occurrences or relation extraction. In our experiments, we find that these methods display different behaviors. Combining them in a learning to rank ensemble successfully improves the performance.

Our results have the following implications. First, they show that combining multiple evidence for ranking related entities can be beneficial. Second, the extraction performance is query-dependent; it would make sense to incorporate the context in which each query entity is mentioned and the types of supporting sentences. Finally, we show that entity networks can be useful for exploring document collections.

Limitations of our approach include the following. First, we do not perform co-reference resolution when extracting entity pairs. Secondly, we extract all pairs of co-occurrences without trying to filter them based on the context of their co-occurrences. Third, we currently ignore the relation type in our relation extraction component.

As to future work, upon analyzing the results, we have discovered common errors related to certain sentence types that affect most methods' performance. Detecting indirect/direct speech as well as enumerations, and automatically filtering them out, is an interesting next step to improve the effectiveness of our approaches. Additionally, to help users of the extracted networks interpret and contextualize the results, we aim to explore the usefulness of automatically linking the newspaper archive from which the networks have been extracted to other archives, similar to [37].

4 Temporal Evidence Classification

We continue with the theme of *entity-entity associations*, but explore another aspect of it: the attributes of entity-entity associations. Associations between a pair of entities can be enriched with optional attributes detailing the nature of the relation. Some relations between pairs of entities in the real world do not hold permanently. Thus, we turn our attention to the extraction of the temporal extent as the attribute of interest for such relations. The setting in this chapter is that the relations between pairs of entities and their relation types are known. We aim to learn the temporal extent of such relations from a document collection.

Temporal relation extraction is the problem of extracting the temporal extent of relations between entities. A typical solution to the temporal relation extraction problem has three main components: (1) *passage retrieval*, (2) *temporal evidence classification*, and (3) *temporal evidence aggregation.* A community-based effort to evaluate temporal relation extraction was introduced in 2011 as a TAC Knowledge Base Population task: Temporal Slot Filling, or TSF for short [112].

An illustration of temporal slot filling is as follows. Having identified a `per:spouse` relation between two entities (Freeman Dyson, Imme Dyson), a system must establish the temporal boundaries from its supporting sentence. In the case of the sentence "*In 1958, he married Imme Dyson*", the goal is to find that the relation lasts from 1958 until the present day. Within the TSF setting, the boundaries are represented as beginning and ending intervals in a tuple (T_1, T_2, T_3, T_4) instead of an exact time expression, so as to allow uncertainty in the system output, where (T_1, T_2) is the beginning interval of the relations, and (T_3, T_4) the ending interval. We investigate temporal relation extraction following this setting. We focus on the temporal evidence classification part, and ask the following question:

RQ2 How can we effectively classify temporal evidence of entity relations?

One of the challenges with relation extraction is the limited amount of training data available to capture the variations in a target corpus: temporal relation extraction faces the same challenge. Employing distant supervision [160] is a way to address the challenge. But generating example training data in the temporal setting is not straightforward: we have to find not only the query and related entity, but also the time expression, in a single text segment.

Employing distant supervision for temporal evidence classification will introduce noise, in the form of labels and additional contexts (e.g., lexical features). A lot of previous work in distant supervision has been dedicated to reducing noise in distant supervision [39, 194, 245]. We are interested in another phenomenon: the class distributions found in training data generated by a distant supervision approach. These distributions become an issue if the distant supervision corpus has a different structure and different characteristics compared to the target corpus, e.g., Wikipedia vs. news articles. We observe that in the case of temporal evidence, news articles and Wikipedia do indeed contain different class distributions; news articles tend to have more current events while Wikipedia articles describe past events. Our working hypothesis is that incorporating prior information about temporal class distribution helps improve our distant supervision approach. We test this hypothesis by comparing a distant supervision strategy with class priors to a distant supervision without class priors. We also demonstrate the effectiveness of our method by contrasting it with a purely supervised approach. In addition, we investigate how the difference in performance in temporal evidence classification affects the final score obtained in the overall end-to-end task.

We discuss related work in Section 4.1. In Section 4.2, we describe our distant supervision approach for temporal evidence classification. Our experimental setup is detailed in Section 4.3. We follow with results in Section 4.4 and conclusion in Section 4.5.

4.1 Related Work

We discuss two groups of related work: on temporal slot filling and on distant supervision.

4.1.1 Temporal slot filling

Some previous work on temporal slot filling uses a pattern-based approach [41]; patterns are defined in terms of query entity, temporal expression, and slot value. For example, the word *divorce* should trigger that the relation *per:spouse* is ending. Other work uses temporal linking between time expressions and events in an event-based approach [40], where the source documents are annotated with TimeML event annotations [180]; the authors use intra-sentence event-time links, and inter-sentence event-event links, following a TempEval approach [229]. Garrido et al. [85] use a graph-based document representation; they convert document context to a graph representation and use TARSQI to determine links between time expressions and events in documents and later map the resulting links into five temporal classes.

Li et al. [130] combine flat and structured approaches to perform temporal classification. Their approach relies on a custom SVM kernel designed around flat (window and shallow dependency) features and structured (dependency path) features. The structured approach is designed to overcome the long context problem. They use a distant supervision approach for the temporal classification part, obtained on Freebase relations. They further extend their approach with self-training and relabeling [113].

Finally, Surdeanu et al. [213] use n-grams around temporal expressions to train a distant supervision system. To be able to use Freebase facts, they find example sentences in Wikipedia, and use a window of five words from the temporal expression, using Freebase facts as *start* and *end* trigger. They use Jaccard correlation between n-grams to determine the association to *start* and *end*. Sil and Cucerzan [205] perform distant supervision using facts obtained from Wikipedia infoboxes. From Wikipedia infoboxes, they retrieve the relevant sentences and build n-gram language models of the relations. In a slightly different setting (exploratory search), Reinanda et al. [185] establish the temporal extent of entity associations simply by looking at their co-occurrence within documents in the corpus.

Our approach to temporal evidence classification differs from most existing approaches in its distant supervision scheme. We use distant supervision to directly perform a multi-class classification of temporal evidence against the five main temporal classes (including the *before* and *after* class), where most of the previous systems train a model to detect the beginning and ending of relationships only.

4.1.2 Reducing noise in distant supervision

With distant supervision [160], indirect examples in the form of relations from a knowledge base such as Freebase and DBPedia are used. From these relation tuples, instances of relations in the form of sentences in the corpus are searched. Text features are later extracted from these sentences that are then used to train classifiers that can identify relations in the text corpus.

Reducing noise is an important ingredient when working with a distant supervision assumption. Relabeling is one such approach; Tamang and Ji [217] perform relabeling based on semi-supervised lasso regression to reduce incorrect labeling. Xu et al. [245] show that instances may be labeled incorrectly due to the knowledge base being incomplete. They propose to overcome the problem of incomplete knowledge bases for distant supervision through passage retrieval model with relation extraction.

Ritter et al. [197] focus on the issue of missing data for texts that contain rare entities that do not exist in the original knowledge base. Riedel et al. [194] work with a relaxed distant supervision assumption; they design a factor graph to explicitly model whether two entities are related, and later train this model with a semi-supervised constraint-driven algorithm; they achieve a 31 percent error reduction.

Bunescu and Mooney [39] introduce multiple instance learning to handle the weak confidence in the assigned label. They divide the instances into a positive bag (at least one positive example) and a negative bag (all negative examples). They design a custom kernel to work with this weaker form of supervision. Surdeanu et al. [214] operate on the same principle, but model the relation between entities and relation classes using graphical models. Hoffmann et al. [105] also use multi-instance learning, but focus on overlapping relations.

What we add on top of existing work is the use of sampling techniques to correct for skewed distributions introduced through distant examples. We propose prior sampling, correcting the distributions of the classes in the generated examples to fit the target corpora.

4.2 Method

The temporal slot filling task is defined as follows: given a relation $R = (q, r, s)$, where q is a query entity, r is a related entity, and s is a slot type, one must find T_R, a tuple of four dates (T_1, T_2, T_3, T_4) where R holds, where T_1 and T_2 form the beginning interval of the relation, and T_3 and T_4 is the ending interval. A system first must retrieve all passages or sentences expressing the relation between q and r. Each sentence and any time information within it will serve as intermediate evidence. This temporal evidence will later be aggregated and converted to tuple representation T_R.

In this chapter, we focus on *temporal evidence classification*. That is, assuming the passage retrieval component has retrieved the relevant passages as intermediate evidence of temporal relations, we must classify whether the time expression t in the passage belongs to one these classes: BEGINNING, ENDING, BEFORE, AFTER, and WITHIN. In the training and evaluation data available to us, only the offsets of the time expression within the document are given for each intermediate evidence, therefore we first extract the paragraph and find the context sentence mentioning t.

Distant supervision for temporal classification The temporal slot filling task, as specified by TAC-KBP, defines 7 types of temporal-intensive relations, i.e., *relations that are fluent and often require temporal specifications*. In our distant supervision approach, we use a separate knowledge base to find instances of the equivalent relations. We use Freebase as our reference knowledge base. That is, we use the temporal information found in Freebase to generate training examples. We manually map the TAC-KBP's 8 temporal relations into 6 Freebase mediator relations. The complete mapping of the relations can be found in Table 4.1.

Table 4.1: Relation mapping to Freebase.

TAC Relations	**Freebase Relations**
per:spouse	marriage
per:title	employment-tenure, goverment-position-held
per:employee-of	employment-tenure
per:member-of	political-party-tenure
per:cities-of-residence	places-lived
per:stateorprovinces-of-residence	places-lived
per:countries-of-residence	places-lived
org:top-employees/members	organization-leadership

In an article, entities and time expressions are not always referred to using their full mentions within a single sentence. Sometimes information is scattered around several sentences: the query entity q in the first sentence, later referred to using a pronoun in the second sentence that contains a time expression, etc. One common way to deal with this problem is to run full co-reference resolution, therefore ensuring all mentions are resolved. We handle this problem by relaxing the distant supervision rule. Rather than retrieving sentences, we retrieve passages containing the query entity q, and related entity r instead. We later replace every pronoun found within the passage with q. Based on our analysis of the Wikipedia articles, this simple heuristic should work, because most Wikipedia articles are entity-centric, and a lot of the pronouns mentioned in the articles will refer to the query entity q.

Each relation that we mapped from Freebase has temporal boundaries *from* and *to*. Following [130], we use Algorithm 1 to generate the training examples, but adapt it to suit to our assumption.

Algorithm 1 GenerateTraining(D, q, r, from, to).

```
Input: Document collection: $D$, query entity $q$, target entity $r$, temporal beginning: $from$, temporal Ending: $to$
Output: Labeled training examples
Retrieve the article of the query entity $q$
Split article into passages
Retrieve passages containing $q$, $r$
Extract all time expressions from the passages
for each $t \in TimeExpressions$ do
    Retrieve the context sentence $s$ containing $t$
    If $t$ is $from$ : label $(s, t)$ as BEGINNING
    If $t$ is $to$ : label $(s, t)$ as ENDING
    If $t$ before $from$ : label $(s, t)$ as BEFORE
    If $t$ after $to$ : label $(s, t)$ as AFTER
    If $t$ between $from$ and $to$ : label $(s, t)$ as WITHIN
```

Sampling the DS examples We manually compared our main corpus (TAC document collection) and our distant supervision corpus (Wikipedia) and noticed some discrepancies. The main corpus mainly consists of newswire articles; one of the main differences between Wikipedia articles and newswire articles is that Wikipedia articles mainly consist of milestone events. In terms of class distribution, this means that most of the generated examples will be in the form of BEGINNING and ENDING class, followed by the BEFORE and AFTER class, with the smallest number of examples belonging to the WITHIN class. In newswire, however, we tend to see something different; most of the time expressions will belong to the WITHIN class.

We argue that using the training data with a “smarter” prior is important. More data not only means more information, but may also mean more noise. This is particularly important with the *relaxed distant supervision* assumption that we have. Therefore, we choose to sample instead of using all of the generated training examples.

We employ two sampling strategies: *uniform*, sampling from our generated training data and deliberately fitting them to a uniform distribution; and *prior-sampling*, where we deliberately construct training data to fit a prior distribution. One way to estimate such a prior is by looking at the distributions of classes in the gold-standard training data that we have. In the case where gold-standard data is not available, we can use a heuristic to estimate the distributions of temporal classes based on domain knowledge or on observations of the target corpora.

In summary, we generate the final training data according to the following steps. First, generate training data with the DS approach described before. Next, estimate class distributions from the (supervised) training data. Then, sample examples from the generated DS data with the probability estimated from the supervised training data (i.e., the empirical prior). Keep sampling the training examples until we reach the target percentage of the DS data. Finally, use the sampled training data to train the multi-class classifier.

Feature representation Both for the training, evaluation, and DS data, we extract context sentences, i.e, the sentences containing the relation and time expression t.

We normalize the context sentences as follows. First, we detect named entities within the sentence and replace the mentions with their entity types (PERSON, ORGANIZATION, or LOCATION). Second, we detect other time expressions within the context and normalize them with regard to the main time expression t, i.e., by normalizing them into TIME-LT and TIME-GT. The idea is to capture the relationships between time expressions as features.

We extract lexical features from normalized sentences. This comprises tokens surrounding the query entity, related entity (slot filler), and time expression. We consider the following four models as our feature representations:

Model-1: bag-of-words All tokens within the normalized sentences are used as features.

Model-2: context window All tokens within the proximity of 3 tokens from the query entity, related entity, and time expression are used as features.

Model-3: context window with trigger words lexicon All tokens within the proximity of 3 token from the query entity, related entity, and time expression are used as features. In addition, a list of keywords that might indicate the beginning and ending of relationships are used as gazetteer features. This list of keywords are expanded by using WordNet to extract related terms.

Model-4: context window with position All tokens within the proximity of 3 tokens from the query entity, related entity, and time expression are used as features. Rather than simply considering them as bag-of-words tokens, the positions of word occurrences are now taken into account as features.

4.3 Experimental Setup

We introduce the dataset and the setup of our experiments. Before that we formulate our research questions as these dictate our further choices.

Research questions We ask the following question in the beginning of this chapter:

RQ2 How can we effectively classify temporal evidence of entity relations?

We expand RQ2 into following research questions to guide our experiments:

RQ2.1 How does a purely supervised approach with various features and learning algorithms perform on the task of temporal evidence classification?

RQ2.2 How does the performance of a distant supervision approach compare to that of a supervised learning approach on the task of temporal evidence classification?

RQ2.3 How does the performance of a prior-informed distant supervision approach compare to that of a basic distant supervision approach on the task of temporal evidence classification?

RQ2.4 How do the approaches listed above compare in terms of their performance on the end-to-end temporal relation extraction task?

Corpora and knowledge base We use the TAC 2011 document collection, which contains 1.7M documents, consisting of news wires, web texts, broadcast news, and broadcast conversation. We use a version of Freebase (dated October 2013) as our knowledge base and retrieve the latest version of Wikipedia as our distant supervision corpus.

Ground truth We use the TAC-KBP 2011 Temporal Slot Filling Task dataset [112] as the ground truth in our experiments. The ground truth comes in two forms: intermediate evidence (with classification labels) and tuples (boundaries of each relation). We use the intermediate evidence to evaluate our temporal evidence classification framework. We later use the provided tuples to evaluate the end-to-end result.

The dataset contains 173 examples in the training set and 757 examples in the evaluation set. The distribution of the classes is shown in Table 4.2.

Table 4.2: Class distribution statistics.

Class	Training	Evaluation	DS Training
WITHIN	66	357	6,129
BEGINNING	59	217	22,508
ENDING	30	110	16,775
BEFORE	9	45	24,932
AFTER	9	28	12,499

Evaluation metric We use F1 as the main evaluation metric for the temporal evidence classification task. For the end-to-end temporal information extraction task, we use the evaluation metric proposed in TAC-KBP 2011, i.e., the Q score. Given a relation r and the ground truth interval tuple G_r, $Q(T_r)$, the quality score of a tuple T_r returned by system S is computed as follows:

$$Q(T_r) = \frac{1}{4} \sum_{i=1}^{4} \frac{1}{1+d_i},$$

where d_i is the absolute difference between T_i in system response and the ground truth tuple G_i (measured in years). To obtain an overall system Q score, we average the Q scores obtained from each relation tuple returned.

In our experiments we test for statistical significance using a paired t-test, indicating significantly better or worse results at the $p < 0.01$ level with ▲ and ▼ respectively.

Experiments We run four contrastive experiments. In Experiment 1, we contrast the performance on the temporal evidence classification task of the different choices for our supervised methods (Model-1, -2, -3, -4), using either Support Vector Machine, Naive Bayes, Random Forest, or Gradient Boosted Regression Tree. In Experiment 2 we examine our distant supervision method and contrast its performance with the supervised methods from Experiment 1. In Experiment 3, we contrast different sampling methods for our distant supervision method.

In Experiment 4 we consider the overall performance on the temporal relation extraction task of our methods; in this experiment we use three "oracle runs" that we have not introduced yet: first, the *Label-Oracle* run uses the actual temporal classification label from the ground truth, use these ground truth label to aggregate the evidence and create the temporal tuples, and compute the end-to-end score; second, *Within-Oracle* assigns all temporal evidence to the WITHIN class; third, *Nil-Baseline* is a lower-bound run that assigns NIL to every element of the temporal tuples.

We use the implementations of the learning algorithms in the Scikit-learn machine learning package [175].

4.4 Results and Discussion

We present the outcomes of the four experiments specified in the previous section.

4.4.1 Preliminary experiment

To answer **RQ2.1**, *How does the performance of the supervised learning approaches on the temporal evidence classification task vary with different representations and learning algorithms?*, we start with a preliminary experiment. The aim of this experiment is to get an idea of the classification performance with a purely supervised approach. The results are shown in Table 4.3.

Table 4.3: Experiment 1. Supervised approaches to temporal evidence classification. Significance of each result is tested against its respective Model-1.

Model	**SVM**	**NB**	**RF**	**GBRT**
Model-1	0.405	0.361	0.402	0.422
Model-2	0.409	0.417▲	0.354▼	0.420
Model-3	0.412	0.418▲	0.361	0.420
Model-4	**0.426**▲	0.424▲	0.241▼	0.422

As shown in Table 4.3, Model-4 with the SVM and NB classifiers achieves the best overall performance. There seems to be a gradual increase in performance from the simpler to the more complex model with SVM and NB classifiers, with the exception of RF. Interestingly, GBRT seems only slightly affected by the different choice of model in this supervised setting.

4.4.2 Distant supervision experiments

Next, we evaluate the distant supervision approach. We aim to answer **RQ2.2**, *How does the performance of the distant supervision approach compare to that of the supervised learning approach?* We generate training examples with the approach described in Section 4.2, and use the full generated training data to train SVM and Naive Bayes classifiers with the same representation models that we use in the previous experiments. The results are shown in Table 4.4.

Table 4.4: Experiment 2 and 3. Supervised, distant supervision, and distant supervision with sampling approaches to temporal evidence classification. Significance of each DS result is tested against its non-resampled variant (column 2).

Model	Supervised	DS	DS-uniform	DS-prior
Model-1 SVM	0.405	0.212	0.379$^\blacktriangle$	0.408$^\blacktriangle$
Model-2 SVM	0.409	0.185	0.389$^\blacktriangle$	0.450$^\blacktriangle$
Model-3 SVM	0.412	0.183	0.384$^\blacktriangle$	0.452$^\blacktriangle$
Model-4 SVM	**0.426**	0.200	0.400$^\blacktriangle$	0.463$^\blacktriangle$
Model-1 NB	0.361	**0.413**	0.379	0.431$^\blacktriangle$
Model-2 NB	0.417	0.299	0.372$^\blacktriangle$	0.451$^\blacktriangle$
Model-3 NB	0.418	0.300	0.368$^\blacktriangle$	0.446$^\blacktriangle$
Model-4 NB	0.424	0.270	0.400$^\blacktriangle$	**0.486**$^\blacktriangle$
Model-1 RF	0.402	0.162	**0.406**$^\blacktriangle$	0.397$^\blacktriangle$
Model-2 RF	0.354	0.177	0.399$^\blacktriangle$	0.418$^\blacktriangle$
Model-3 RF	0.361	0.176	0.391$^\blacktriangle$	0.403$^\blacktriangle$
Model-4 RF	0.241	0.171	0.399$^\blacktriangle$	0.446$^\blacktriangle$
Model-1 GBRT	0.422	0.142	0.316$^\blacktriangle$	0.344$^\blacktriangle$
Model-2 GBRT	0.420	0.137	0.343$^\blacktriangle$	0.418$^\blacktriangle$
Model-3 GBRT	0.420	0.138	0.343$^\blacktriangle$	0.403$^\blacktriangle$
Model-4 GBRT	0.422	0.140	0.399$^\blacktriangle$	0.433$^\blacktriangle$

We observe that the distant supervision approach trained on the full set of generated examples (the column labeled "DS") performs poorly, well below the supervised approach. We hypothesize that the accuracy drops due to the amount of noise generated with our distant supervision assumption trained from full data, and different class distribution statistics.

In Section 4.2, we proposed our prior-sampling approach for distant supervision. The next experiment is meant to answer **RQ2.3**, *How does the performance of our prior-informed distant supervision approach compare to that of the basic distant supervision* *approaches?* We sample 20 percent of the generated examples datasets with the following strategies: *uniform* and *prior*. The results are also shown in Table 4.4, in the columns labeled "DS-uniform" and "DS-prior," respectively.

By observing the results in Table 4.4, we notice that distant supervision with prior sampling performs the best, for every combination of model and classification method. *Uniform* sampling already helps in improving the performance, and prior sampling successfully boosts the performance of the basic distant supervision (for all four models) further. Distant supervision with prior sampling also performs consistently better than the supervised approaches (Table 4.3) in many cases—interestingly, for GBRT, DS-prior only outperforms the supervised methods with sufficiently complex feature representation (Model-4 GBRT).

4.4.3 End-to-end experiments

Next, we answer **RQ2.4**. That is, we consider how the classification performance on temporal evidence classification affects the end-to-end result. We take the best performing models from the previous experiments and evaluate their end-to-end scores. The results are shown in Table 4.5.[1]

Table 4.5: Experiment 4. End-to-end scores (Avg-Q) next to F1 scores for temporal evidence classification. Significance of each supervised and DS result is tested against the best supervised method (row 4).

Model	**Avg-Q**	**F1**
Label-Oracle	0.925	1.000
Within-Oracle	0.676	0.302
Nil-Baseline	0.393	N/A
Supervised		
Model-4 SVM	0.657	0.426
Model-4 NB	0.648	0.424
Model-4 RF	0.573▼	0.241▼
Model-4 GBRT	0.649	0.422
Distant supervision		
Model-4 SVM	0.669▲	0.463▲
Model-4 NB	0.679▲	0.486▲
Model-4 RF	0.653	0.446
Model-4 GBRT	0.669▲	0.433

From Table 4.5, we see that Model-4 RF (F1 on temporal evidence classification 0.446) and Model-4 GBRT (F1 on temporal evidence classification 0.433) translate into 0.653 and 0.669, respectively, in terms of Q-score. This means that the misclassifications that Model-4 RF produces have a larger impact than those of Model-4 GBRT. However, the difference in performance is not large.

[1] As the Nil-Baseline is applied directly to the final tuples rather than the classification labels, there are is no F1 score for this run.

The evaluation of this end-to-end task is important because not every misclassification has a similar cost. Misclassification of class A into class B can result in a huge increase/decrease in performance. First, the classification performance does not directly map to the end-to-end score. Second, several relations have more pieces of evidence than others; performing misclassifications on relations that have a lot of supporting evidence would probably have less effect on the final score.

The state of the art performance, using distant supervision [130], achieves an end-to-end Avg-Q score of 0.678 (on training data), where we achieve 0.679 (on evaluation data). However, our scores are not directly comparable since we reduce the number of classes (and the amount of evidence) in our evaluation. It is important to note that [130] use a complex combination of flat and structured features as well as the web, where we use relatively simple features with Wikipedia and prior sampling.

Furthermore, our approach manages to achieve the same level of end-to-end performance as the Within-Oracle run, while achieving a significantly better F-score. More pieces of evidence were actually classified correctly, though this was not reflected directly in the end-to-end score due to issues described above.

4.4.4 Error analysis

We proceed to analyse parts of our end-to-end results to see what is causing errors in the temporal evidence classification task. We found several common problems.

Semantic inference Some problems had to do with the fact that several snippets require semantic inference. The fact that someone dies effectively ends any relationships that this person had. Another example is when someone marries someone (A marries C), and this beginning of relationships effectively means the end of relationships for previous relations (A and B). A more complex method to deal with this type of semantic inference is needed, simple classification does not work so well. Here is an example:

> *Angela Merkel is married to Joachim Sauer, a professor of chemistry at Berlin's Humboldt University, since 1998. Divorced from Ulrich Merkel. No children.*

For this example the fact is that the time expression 1998 happens *after* with regard to the *spouse* relation between Angela Merkel and Ulrich Merkel.

Concise temporal representations Newspaper articles contain lots of temporal information in a concise way. For example in the form (X–Y). This implicit interval range is not expressed in a lexical context but rather with symbolic conventions. In several articles, the information encoded is almost tabular rather than expressed explicitly. For example:

> *Elected as german chancellor Nov. 22, 2005. Chairwoman, christian democratic union, 2000-present. Chairwoman, christian democratic parliamentary group, 2002–2005.*

Complex co-reference Named and pronoun co-reference can probably still be handled with heuristics, but phrase-based co-references like *the former president* are hard to resolve.

> *The former prime minister (1998-2001) is once again angling for the top job after taking over as chairman of the Labor Party and being appointed minister of defense in Ehud Olmert's government in 2007.*

Complex time-inference BEFORE and AFTER are especially tricky to deal with because they require additional inference. Even if a passage contains the word *after*, the time expression linked to it would probably contain the *before* relation.

> *He was called up by the Army in the spring of 1944, after marrying bea silverman in 1943, and was sent to The Philippines.*

For the above example, 1943 happens *before* the "person joined the Army" event.

We observe quite a number of these cases on the evaluation data. Furthermore, the lack of context on some examples and evidence that is scattered around multiple sentences complicates the problem even more. Because of semantic and implicit evidence, temporal evidence classification remains a challenging task. In order to achieve a better absolute performance, collective classification/inference of evidence seems an interesting option.

Relation mismatch We find that the evaluation set contains relations that are not mentioned in the task description. For instance, the *person:schools-attended* and *organization:subsidiary* relations are not within the seven type of relations described in the task description. This inclusion especially hurt the performance of a distant supervision approach, because we did not map any Freebase relations from the *schools-attended relation* and generate training examples. We noticed the distant supervision approach performs poorly on this type of relations. Meanwhile, the purely supervised approach can cope well with these relations exist in the training data.

> *Born in Prague on June 19 1941, Klaus graduated from the capital's University of Economics in 1963 and was afterwards permitted the rare privilege at the time of training courses in Italy and the US.*

4.5 Conclusion

In this chapter, we have considered an important aspect of entity-entity associations: their attributes. In particular, we have focused on the temporal attribute, which is important in relations that do not hold permanently. The extraction of temporal attributes consists of three step: (1) *passage retrieval*, (2) *temporal evidence classification*, and (3) *temporal evidence aggregation* . Focusing on the evidence classification part, we have asked the following question:

RQ2 How can we effectively classify temporal evidence of entity relations?

To answer RQ2, we experimented with a various setting of distant supervision approaches, and also supervised approach. We have presented a distant-supervision approach to temporal evidence classification. The main feature of our distant supervision approach is that we consider the prior distribution of classes in the target domain in order to better match the label distributions of distant supervision and target corpora.

We have shown that our prior-informed distant supervision approach outperforms a purely supervised approach. Our method also achieves state-of-the-art performance on end-to-end temporal relation extraction with fewer and simpler features than previous work. We have also considered the contribution of our temporal evidence classification component to the performance on the overall temporal relation extraction task.

Our findings have the following implications. First, we show the importance of distribution matching for distant supervision; both in the context of temporal evidence classification, and for relation extraction in general. Secondly, we show that the evidence classification performance does not always translate directly to the end-to-end score; more investigation into this relationship would be beneficial. Finally, incorporating more distant supervision examples does not always mean improved performance, as they tend to bring more noise.

There are also some limitations to our work. First, we rely on simple lexical features for temporal evidence classification. Second, our method normalizes an entity mention with fairly simple heuristics. Third, our temporal expression extraction accuracy is limited by the library that we use, some noise might be introduced because of this.

Our error analysis on the temporal evidence classification task revealed several issues that inform our future work aimed at further improving the performance on the subtask of temporal evidence classification, and the overall temporal relation extraction task. In particular, we intend to deal with the challenging aspect of semantic inference over relations found in the evidence passage. Another interesting direction that we aim to tackle is dealing with evidence that is scattered across multiple sentences.

5
Impact-based Entity Recommendations from Knowledge Graphs

In this last chapter on the theme of *entity-entity associations*, we move beyond document collections and consider an entity-oriented search scenario on structured data, i.e., knowledge graphs. In contrast to work on ranking related entities introduced in Chapter 3, we now bring the nature of the entity relationships into the forefront. Specifically, we study how direct connections and indirect connections (i.e., built from a sequence of direct relations) in the knowledge graph affect a specific recommendation goal that we aim to address. In this setting, we focus on entity relations data from two domains: *politics* and *business*, and develop methods to support users from these domains in analysis and decision-making.

Information about entities and their connections—often encoded in knowledge graphs—is appearing ubiquitously in the context of modern search engines [136, 178]. In a Web setting, knowledge graphs are particularly useful for query understanding, presenting entity summaries, and providing explanations for search results [24, 94, 234]. Another popular application of knowledge graphs is to power entity recommendations. Existing work in this area mostly focuses on the Web search domain, in which the main features of the recommendation algorithm are typically based on behavioral signals extracted from users' search sessions [20, 23, 119, 157]. Entity recommendations that are generated and scored primarily from the semantics of the connections between entities in a knowledge graph are less well-studied.

In this chapter, we consider the task of *impact-based entity recommendations* from knowledge graphs: recommending entities with respect to a query entity based on impact. We define *impact* as tangible effects or consequences of any major event involving the query entity to its related entities. As an illustration, consider the following use case. Suppose we have a knowledge graph containing company, place, and person entities, and several relationship types connecting these entities. Consider an event such as "*a change of management in Walmart*", where the query entity e_q is *Walmart*. Assuming there are connections from *Walmart* to a number of companyand person entities in the knowledge graph, which of these entities will be affected the most? The subgraph between the two entities S_{uv} contains all simple paths (i.e., no repeated nodes and no loops) connecting the entities in less than k hops. *Walmart* and the entities in this subgraph (e.g., *Politician-A, Politician-B, Lobbyist-C*) can be connected by a multitude of paths comprising different relationship types. Suppose *Politican-A* received donations from *Walmart*, and *Lobbyist-C* works directly with *Walmart*, then both *Politician-A* and *Lobbyist-C* will be affected, i.e., the event in *Walmart* has a strong impact on *Politician-A* and *Lobbyist-C*. Meanwhile, *Politician-B*, who has an indirect connection due to a family member who works there will less likely be affected, i.e., receiving a lesser impact.

Working in the setting described above, we will run into several challenges. For one, knowledge graphs are inherently heterogeneous, i.e., they contain multiple types of entities and multiple types of relationships between entities. Learning which relationship types are important for impact prediction is already challenging. Moreover, when multiple links are combined into a path, the number of possible link combinations can grow exponentially. Finally, we need a way to aggregate the impact from a query entity to the related entities in the case of multiple paths. Most of the work on heterogeneous graphs considers a limited number of relationship types [8, 126, 176]. In this chapter, we are particularly interested in learning the impact-based recommendations on highly-heterogeneous graphs, which requires a different strategy. We ask the following question:

RQ3 Given graph-based information of entity relations with types, can we effectively recommend related entities based on their direct and indirect connections to a query entity?

To address this question, we propose two novel methods for the impact-based entity recommendation task. Our first approach is based on learning to rank, in which we extract features from the subgraphs connecting the query entity e_q and related entity e. The intuition is that we can leverage signals such as the path length between entities in combination with other features such as the different relationship types in the subgraph to predict impact. Our second approach is inspired by Bayesian networks, in which we explicitly model the propagation of impact in the knowledge graphs in a probabilistic manner and learn to deal with different relationship types accordingly. We make intermediate predictions from the query entity to intermediate entities at every stage, i.e., making predictions locally, and propagate this prediction to the related entity. In the learning phase, we optimize the weights of each relation type globally within this propagation sequence, taking into account all possible paths. Our approach is unique in the sense that it utilizes shared parameters of conditional probability by relationship type across all the subgraphs.

Our main contributions can be summarized as follows. First, we introduce the novel task of *impact-based entity recommendations* from knowledge graphs. Second, we propose two approaches for entity recommendations in this setting. Third, we perform an in-depth analysis and compare our methods against a strong baseline for entity recommendations.

The remainder of this chapter is organized as follows. We define the task and setting in §5.1. We describe our approaches to impact-based entity recommendations from knowledge graphs in §5.2. A detailed description of our experiments and the data used is given in §5.3. We discuss the results of our experiments in §5.4 and conclude in §5.5. Table 5.1 details the main notation that we use throughout the chapter.

Table 5.1: Glossary of the main notation used in this chapter.

Symbol	Gloss
KG	a knowledge graph
e	an entity, where e_q is the query entity
l_{uv}	a directed edge/link connecting entity u and v
S	a subgraph of KG
$S_{e_q e}$	a subgraph containing the set of all paths connecting query entity e_q and target entity e
Ψ	a learning to rank model
$\phi_{S_{uv}}$	features extracted from paths S_{uv}
$B_{S_{uv}}$	a belief graph derived from the subgraph built from all paths S_{uv}
E	a random variable representing node e in B
Q	a random variable representing node e_q in B
I	a random variable representing node i between e_q and e in the belief graph B
$P(E\|Q)$	the impact probability of node E given Q
$\phi_{l_{uv}}$	features extracted from directed edge e_{uv}
ω	a conditional probability function i.e., $P(E\|D)$ for directly connected entity nodes E and D
Ω	aggregated prediction function to estimate $P(E\|Q)$, used in learning and inference

5.1 Problem Formulation

Recall that our primary goal in this chapter is to develop a method for impact-based entity recommendations from knowledge graphs. We formally define the task as follows:

Definition 20 ***Impact-based entity recommendations*** *Given a query entity e_q, rank each entity $e \in KG$ with respect to its predicted impact given a major event to the query entity.*

We formulate a general approach to solve this task as estimating the impact-based relevance between pairs of entities, i.e., $rel(e_q, e)$ and ordering them in a descending order. We employ two approaches to estimate this relevance. One way to estimate the impact-based relevance is by predicting the *relevance* of the entity with respect to the query within a learning to rank framework, which we detail in Section 5.2.1. Second, we can directly estimate $rel(e_q, e)$ as $P(e|e_q)$, the impact probability of an entity e given query entity e_q, through a graph-based inference algorithm as detailed in Section 5.2.2. Finally, as a baseline, we consider a relevance estimation method based on supervised *graph proximity*, which we adapt to support heterogeneous graphs in Section 5.3.3.

In order to reduce the search space for ranking entities $e \in KG$, we first obtain a subgraph of candidate entities, S. This subgraph is fetched from the KG following a traversal procedure: given a query entity e_q, we first retrieve all entities directly related to the query entity and then perform depth-first traversal to retrieve the next set of candidate entities up to k hops. The query entity, related entities, and all the relations between them form a subgraph S_{e_q}.

5.2 Methods

In this section, we detail our two proposed methods for impact-based entity recommendations: (1) a learning to rank-based model and (2) a probabilistic model inspired by Bayesian networks.

5.2.1 Learning to rank

Our first approach does not explicitly consider propagation. Here, we rank each candidate entity $e \in E$ based on the following score:

$$rel(e, e_q) = \Psi(\phi_{e,e_q}), \tag{5.1}$$

where Ψ is a machine learned ranking model that makes predictions based on $\phi_{e_q,e}$, i.e., the feature representation extracted from subgraphs containing all paths Pe_qe connecting related entity e to query entity e_q.

Next we describe our feature representation, the details of which are listed in Table 5.2. We consider length, magnitude, and type features that are meant to capture different intuitions for impact prediction such as the fact that direct and/or multiple-path connections are important, but also the fact that single relationship-type connections are important. Furthermore, these simple intuitions can be combined to form a complex set of rules, encoding the global structure of the subgraph whilst keeping the number of features linear with respect to the relationship types.

Length features This feature group is designed to capture the general characteristics of all paths connecting the query and target entity. In particular, we focus on the length of the paths and summarize this subgraph by extracting the number of paths connecting the two entities (i.e., the number of individual paths in p_e), shortest path length, longest path length, and the average path length.

Magnitude features Relations in a knowledge graph can have additional properties. Consider the following relationship: *campaign-donor* from a company to a person. Naturally, a property like *amount* can be included in the knowledge graph. Some of these properties can be normalized into weights that indicate the strength of a connection. One way to normalize this property into a weight for this particular relation is to divide this quantity against the sum of quantities of all relations originating from the same entity. Another way would be doing the same but then based on the target entity. This feature group therefore aims to capture the strength of the relationship that exists between all paths connecting the two entities. We apply the normalization method described above for each relation property. Then, for a path $p_{qe} \in S_{e_qe}$ connecting query entity q and entity e we consider two different types of aggregations and compute the path magnitude by

Table 5.2: Features used for ranking candidate entities. Features are extracted from all paths connecting the query and related entity. *N* indicates numeric vectors, while *V* indicates vector features.

Feature	**Description**	**Type**
`NumPaths`	Number of paths connecting the two entities	N
`MaxPathLen`	Longest path length	N
`MinPathLen`	Shortest path length	N
`AvgPathLen`	Average path length	N
`MinPathMagnitudeProd`	Minimum path magnitude, aggregated by product	N
`MaxPathMagnitudeProd`	Maximum path magnitude, aggregated by product	N
`AvgPathMagnitudeProd`	Average path magnitude, aggregated by product	N
`MinPathMagnitudeSum`	Minimum path magnitude, aggregated by sum	N
`MaxPathMagnitudeSum`	Maximum path magnitude, aggregated by sum	N
`AvgPathMagnitudeSum`	Average path magnitude, aggregated by sum	N
`BagRelationTypes`	Types of relations in the paths	V
`BagEntityTypes`	Types of entities in the paths	V

aggregating the strength of connections between the two entities as follows:

$$magnitude_{prod}(q, e) = \prod_{l_{uv} \in p_{eq}} weight(l_{uv}), \tag{5.2}$$

where $weight(e)$ indicates the weights of all edges connecting two entities aggregated by multiplying the strength of all intermediate edges. We also consider:

$$magnitude_{sum}(q, e) = \sum_{l_{uv} \in p_{eq}} weight(l_{uv}), \tag{5.3}$$

where $weight(l_{uv})$ indicates the weights of the edge l_{uv} connecting two adjacent entities u and v. Additionally, we compute the maximum, minimum, and average aggregated magnitudes as features as well.

Type features This feature group is designed to capture the type of entities and entity relationships that exist between all paths connecting the two entities. For each entity and relation type, a boolean feature is extracted to indicate whether the particular entity/relation type is found within the paths connecting the two entities. This binary vector that indicates the occurrence of entity and relationship types is then used as a feature.

5.2.2 Impact propagation

Our second approach directly considers the propagation of impact from one node to another, starting with the query entity q. We do so by creating a belief graph $B_{P_{e_q e}}$ based on the knowledge subgraph $S_{e_q e}$ for each query-entity pair (e_q, e) and then performing a propagation-like procedure on this belief network. We design a simple but efficient algorithm inspired by belief propagation algorithms from related work [162, 174]. First, we represent each entity node e as a random variable E indicating the impact on entity e. The links in the graph indicate a causal dependency relationship between entities. However, given the fact (1) that the query and related entities can be connected by multiple paths and (2) how we construct the subgraph, S will have a tree-like structure. Moreover, the knowledge graph links are directed, reflecting a relation triple $\langle d, r, e\rangle$ denoting source entity d, relationship r, and target entity e. Projecting this onto the belief graph B, node D will become the *parent* of node E.

For inference, we simply instantiate the query node Q, assigning it as an observed variable and thus indicating that it is an entity that is affected by an event. We then propagate this state to the other nodes, obtaining the impact probabilities of all other entities in the subgraph. Our extension of the Bayesian network utilizes a *parameterized conditional probability* model that estimates the transitive propagation probability after representing the connection between two adjacent entities l_{uv} as features $\phi(l_{uv})$. In the following subsections, we further detail our approach. We first provide an overview of the inference and prediction procedure and then we detail how we learn the parameters.

Inference

$\Omega(.)$ is the forward propagation function, which applies the conditional probability ω sequentially from the source to target node. The probability of $P(E|Q)$ is reduced to the joint probability $P(Q, E, I)$, where I denotes all intermediate nodes between Q and E. Therefore, following $P(E|Q) = \frac{P(Q,E)}{P(Q)}$, we can compute the conditional probability $P(Q|E)$ as the joint probability $P(Q, E, I_1, .., I_n)$. We further assume *local propagation*, i.e., a node must be affected for it to be able to spread the influence to a neighboring (child) node. With this assumption, any parent node D connected to child node E must be affected, allowing us to avoid computing all the combination of values of all intermediate nodes as the joint probability of the subgraph. We can therefore compute the conditional probability efficiently in a top-down manner, propagating the joint probability from query node Q to entity node E. In this forward propagation, the joint probabilities can be computed recursively as detailed in Algorithm 2. Conditional probabilities are computed for each link by applying $\omega(\phi(l_{uv}))$, which effectively produces solely local predictions. The joint probability is computed incrementally in a top-down fashion, giving us the joint probability $P(Q, I_1, .., I_n, E)$ once we reach the related entity node E.

Learning

In the learning phase, we learn the conditional probability of impact propagation parameterized by the relationship type. One of the key ingredients to perform inference in a Bayesian network are *conditional probabilities*, which serve as the parameters of the model. In a normal Bayesian network, these parameters θ are typically learned from data.

Algorithm 2 computeProb(B, q, e)

```
Input: Belief graph $B$, query $q$, candidate entity $e$
Output: Probability $P$
if IsRoot(E) then
    $P \leftarrow 1.0$
    return $p$
else
    $E \leftarrow getIncomingLinks(E)$
    $C \leftarrow \{\}$
    $M \leftarrow \{\}$
    for $edge \in E$ do
        $c \leftarrow \omega(\phi(edge))$
        $C \leftarrow C \cup c$
        $m \leftarrow computeProb(B, q, edge.src)$
        $M \leftarrow M \cup m$
    $P \leftarrow causalAggregation(C)joint(M)$
    return $P$
```

One of the important benefits of working with knowledge graphs is that these parameter values can be shared throughout the whole network, i.e., we only need to learn a single *conditional probability function* ω that encodes the different conditional probabilities between entities u and v directly connected by edge e. In the following section, we will discuss how we learn this *conditional probability model* from our data.

Instead of learning all values of the parameters $P(E|D)$ for every combination of E and D, we learn a conditional probability model ω through a gradient-descent optimization procedure. The conditional probability model will be shared across different subgraphs generated from our (q, e) pair. In particular, the function *computeProb*, denoted as Ω in Algorithm 3, applies the forward inference procedure that we introduced in the previous subsection. Each link between entity e from its parent d is represented as feature vector $\phi(l_{de})$ and we use the one-hot vector of the relationship type of l and the magnitude of the relation w as the feature vector ϕ. The weight will default to 1.0 if the link does not contain any magnitude information.

The propagation probability $P(E|D)$ between two adjacent entities e and d connected by edge l_{de} is indicated in upper case E and D can be estimated as follows:

$$P(E|D) = \omega(l_{de}) = \frac{1}{1 + e^{\theta\phi(l_{de})}}, \tag{5.4}$$

that is, the probability of entity e given parent entity d can be estimated through a sigmoid function using weights θ and the binary feature vector extracted from edge that connect the two entities.

Our optimization procedure to learn the function ω is detailed in Algorithm 3. During training, the prediction $\Omega(q_m, e_m)$ for each data point m is made by propagating evidence from query to related entity. We first initialize the weights θ of each relationship type in a random fashion. Next, we make local predictions on direct relations based on these randomly initialized weights and compute the predictions for every connected pair of

Algorithm 3 Learning conditional probability model with L-BFGS.

```
Input: Training data points $M$
Output: Conditional probability model: $\omega$;
$\theta \leftarrow initializeWeights$
while $notConverged(w)$ do
    $\mathcal{L} \leftarrow 0.0$
    for each $m \in M$ do
        $p_m \leftarrow \Omega(subgraph(m))$
        $\mathcal{L} \leftarrow loss(y_m, f_m)$
    $\theta \leftarrow updateWeights(\theta, \mathcal{L}))$
return $\omega(\theta)$
```

entities. Then, we propagate these predictions to the child node, and so on. In the event of multiple causes, we deal with an aggregation function that we will detail in the next subsection.

We consider the following loss function:

$$\mathcal{L} = \sum_{i}^{N} \log(1 + e^{-y_i \Omega(x_i)}), \tag{5.5}$$

where y_i is the label for datapoint i converted to probabilities, and $f(x_i)$ the respective probabilistic prediction on data point i.

With the loss function $\mathcal{L}$, we update the weights by its derivative using the L-BFGS algorithm [139]. $\Omega(.)$ gives the prediction at training time using the current parameter values θ. By learning the weights of the relationship through forward inferencing, each relation type is optimized based on its occurrence in the context of other relations within the subgraph.

Turning labels to probabilities Our impact propagation method expects probabilities $P(E|Q)$ as input during training. With relevance labels $g_{e_q e}$ in our training data denoting the relevance between (e_q, e), we convert the labels to probabilities as follows: $P(E|Q) = \frac{g_{e_q e}}{4}$,which divides $label$ by the highest possible label, yielding a value between 0.25 and 1.0.

Causal aggregation

Since there can be multiple links directed at node E, E will have multiple parents. This means that for each entity node with multiple parents the impact contributed by each parent needs to be aggregated and taken into account, which is equivalent to modeling *causal aggregation*. There are different ways to address causal aggregation [174]. With our Bayesian network-like approach, it is not feasible to learn the conditional probabilities with joint causes because: (1) we have multiple belief graphs instead of a single Bayesian network, and (2) it will require a very large amount of data to estimate all the conditional probabilities in these multiple scenarios. To address this issue, we employ the *noisy-or distribution*, which allows us to compress our conditional probability model. Note that, our learning framework is generic and can be extended with other methods for modeling such causal aggregation. The noisy-OR distribution has the property that each possible cause (i.e., parent in the graph) can exercise its influence independently [174], which is what we want, as we want to accumulate effects from multiple paths. We utilize the noisy-OR distribution to combine evidence from multiple parents. This method is computed as follows:

$$P(E|D_1, D_2, ..., D_n) = 1 - \prod_i \big(1 - P(D|D_i)\big), \tag{5.6}$$

where i iterates over all parents of D in the belief graph.

Implementation details

Here we detail some final notes on how to put these components together.

Graph preparation To allow the recursive computation of the forward propagation detailed in Algorithm 2, the belief graph B must not contain any directed loops. We ensure this constraint by inverting the directionality of an edge if for an edge u_{uv} a destination node v is already in the ancestor list of the current node u. We also invert the relation types and weights accordingly for relation types that are not symmetric. For relations that are symmetric, inverting this relation magnitude is not a problem as the edges in our belief graph does not necessarily depict actual causal relationship as in a common Bayesian network, but rather a flow of information.

Feature representation Currently, we utilize the relationship type and magnitude as features, but our framework is generic and can be extended to include additional features, such as entity types, entity attributes, etc. We build the belief graphs $B_{p_{e_qe}}$ for all query-entity pairs, and learn a *shared conditional probability model* ω in a supervised manner from the training data, optimizing against the likelihood of observed labels through gradient descent. We initialize the weights of relation types θ for the ω randomly, and update it until convergence.

5.3 Experimental Setup

In this section, we describe our experimental setup and detail the research questions that drive our experiments, the data and baseline we use, evaluation metrics, and parameter settings of the methods.

We ask the following question:

RQ3 Given graph-based information of entity relations with types, can we effectively recommend related entities based on their direct and indirect connections to a query entity?

Our experiments are driven by the following research questions, derived from RQ3:

RQ3.1 How do our proposed methods and the baseline perform on the task of impact-based entity recommendations?

RQ3.2 How does the impact propagation method compare against the learning to rank method?

RQ3.3 How do our proposed methods perform across different queries?

RQ3.4 Can the impact propagation model learn which relationship types are important for impact-based recommendation?

5.3.1 Data

We perform our experiments on both a publicly available knowledge graph as well as an industrial knowledge graph from a commercial data provider company. Our public dataset is based on the LittleSis database [121], which primarily focuses on the political domain. This knowledge graph contains various object types including *people*, *organizations*, and *locations*. The relationships can be grouped into ten main categories: *position*, *student*, *member*, *relation*, *donation*, *service*, *lobbying*, *professional* and *ownership*. Currently, this knowledge graph contains facts about 100,000 entities. We perform some preprocessing and extract a subgraph of this data, leaving out rare relation types. We end up with 168 relation types which comprise 900,000 relations.

Our industrial dataset is based on a knowledge graph of a commercial data provider company. It contains entities and relationships that covers the business and finance domains. The entities are of multiple types including *companies*, *people*, *locations*, etc. This knowledge graph is highly-heterogeneous, with more than 100 different relation types.

5.3.2 Relevance assessments

For our relevance assessments we generate the candidate related entities using the following procedure. We first perform candidate generation with the traversal algorithm described in Section 5.1. To limit the size of the query subgraph, and the number of candidates, we limit our traversal based on the degree of each node. If a node has an in-degree above a threshold $k = 30$, we will not continue traversing the incoming links, as we assume these to be very general connections.

We sample a number of query entities and extract subgraphs from the knowledge graph by traversing for a maximum of k hops from the query entity. To make sure we have a representative number of entities in each hop, we sample candidate entities separately for each hop, i.e., we compute the shortest distance from a candidate entity to the query entity, and sample entities with shortest distance in $k \in 1, 2, 3$. This ensures that we have a number of direct and indirect candidates in our dataset.

We finally present these query and candidate pairs to assessors to judge the impact-based recommendations. We utilize crowdsourcing to collect our relevance judgments. More specifically, we design a task in which the assessors have to decide the query entity's impact to the candidate entity. We ask the assessors to judge the impact within a 4-grade relevance level. We instruct the crowd to annotate as follows:

- **Not relevant**: an event affecting query entity q will have no impact on the candidate entity e.

- **Somewhat relevant**: an event affecting query entity q might have an impact on the candidate entity e, although this impact might be limited.

- **Relevant**: an event affecting query entity q will have an impact to the candidate entity e.

- **Highly-relevant**: an event affecting query entity q will have an obvious and strong impact on candidate entity e.

We initially judge 60 query-entity pairs and use that as test questions to control the quality of the crowd annotation. We use CrowdFlower as our annotation platform. CrowdFlower will automatically exclude annotators whose agreements fall below a threshold (set to 0.7). In the end, we collect 1600 judgments of query-entity pairs, comprising 54 query entities for the experiments on LittleSis.

5.3.3 Baseline

We compare the performance of our proposed methods: Learning to Rank (LTR) and Impact Propagation (IP) against a baseline based on supervised random walks in our experiments. We detail this baseline in the remainder of this section. Although random walk-based methods are typically used to perform unsupervised recommendations on graph data, we adopt a supervised random walk method based on [8] to incorporate parameterized edge weights. That is, we learn different transition probabilities for each edge type in the knowledge graph. Intuitively, this allows for a better approximation of edge weights. This supervised random walk method learns from pairwise preferences of entity page ranks. For every data point $i, j \in D$ in the training data, we generate paired preference x_i, x_j for every pair that satifies $y_i < y_j$. In the learning step of the PageRank edge weight parameters, we aim to reduce the number of incorrectly ordered preference pairs. More specifically, we aim to optimize the following loss function:

$$\min F(\theta) = ||\theta||^2 + \lambda \sum_{(p_l, p_d) \in P} h(p_l - p_d), \tag{5.7}$$

where θ is the parameter for edge weights, λ the regularization parameter, P is a list of known PageRank ordering such that $p_l < p_d$. $h(.)$ is the loss function computed from the pairwise PageRank difference of any two nodes in the subgraph. Following [8], we choose the Wilcoxon-Mann-Whitney (WMW) loss with b set to 0.5:

$$h(x) = \frac{1}{1 + \exp(-x/b)}, \tag{5.8}$$

which is differentiable and has been proposed to maximize AUC in [247]. Using the relation type to parameterize the transition probabilities of the links, we ultimately learn the transition probability and the respective edge weights for the relation type. We use the learned edge weights to compute page rank scores in the heterogeneous graph and consider the PageRank scores as the final score to rank the entities for recommendation.

Table 5.3: Results on the LittleSis dataset.

Method	P@1	P@3	P@5	P@10	N@1	N@3	N@5	N@10
SRW	.717	.731	.724	.711	.699	.787	.808	.835
LTR	**.836**	**.816****	**.790**	**.759***	**.798***	**.841****	**.858***	**.874****
IP	.750*	.786	.752	.751*	.736*	.817	.825*	.854**

5.3.4 Metrics and significance testing

We evaluate the proposed approaches in a rank-based setting. As our problem can be considered as a form of entity recommendation, we use metrics commonly used in document retrieval: precision at m and nDCG@m where $m \in 1, 3, 5, 10$. We compute DCG using: $DCG@k = \sum_{i=i}^{k} \frac{2^{rel_i}-1}{\log_2(i+1)}$,and normalize DCG with the ideal DCG to obtain nDCG. To determine whether the difference in performance between methods is statistically significant we apply the student's paired t-test and use * to denote $p < 0.1$ and ** for $p < 0.05$.

5.4 Results and Discussion

In this section we present the results of our experiments, answer the research questions we pose in the previous section, and discuss the insights that we gained. We first turn to answering **RQ3.1** and **RQ3.2** by performing experiments on the LittleSis and industrial data.

Table 5.3 details the results of our experiments on the LittleSis data. Overall, the learning to rank method obtains the best performance both in terms of precision and NDCG. We further observe that both our methods improve upon the supervised random walk baseline in all metrics. We look at the performance of the learning to rank approach on the public dataset. LTR obtains a 10% improvement over the baseline in terms of P@3 and 5.6% in P@10. In terms of NDCG@10, LTR achieves a 4.8% improvement. When we turn to the performance of the impact propagation (IP) approach we find that it obtains a 7% improvement over the baseline in terms of P@10. In terms of NDCG@10, IP achieves a 4% improvement over the baseline.

When we compare the performance of the methods on the industrial knowledge graph, similar trends emerge. First, we find that both our proposed methods improve upon the supervised random walk baseline on almost all metrics and settings.[1] This improvement is significant, with the LTR method again achieving the best overall performance in terms of precision and NDCG. We do note that the impact propagation method also improves significantly over the baseline. The improvements of LTR and IP are also of greater magnitude than on the LittleSis dataset.

Finally, we summarize the results of this experiment. We observe that the learning to rank approach tends to achieve the most improvements in terms of precision, outperforming the baseline and IP. Both methods consistently obtain significant improvements over the baseline on the public and industrial datasets.

[1] Due to the proprietary nature of this dataset we cannot publish any absolute scores, unfortunately.

In summary, both our proposed methods show their potential for impact-based entity recommendations, obtaining improvements both on the LittleSis data and on the industrial data. Overall, the learning to rank method obtains more improvements compared to the impact propagation method.

5.4.1 Performance across queries

Here we answer **RQ3.3**, comparing the performance of the different method across queries to gain more insights.

Win-loss analysis First, we want to discover whether LTR achieves the overall improvements while consistently outperforming IP. We do so by comparing the NDCG@10 and P@10 of LTR and IP across all queries.

Table 5.4 shows a detailed result of this contrastive analysis. In terms of NDCG, we observe that IP performs better than LTR on 13 queries, while LTR wins on 28 queries. There are 13 occasions where the performance ends up in a tie, which means they achieve the same scores. When it comes to precision, we observe that IP performs better than LTR on 14 queries, while LTR wins on 18 queries. There are 18 occasions where the performance ends up in ties. This result indicates that the two methods perform differently on different circumstances.

Performance on difficult queries Our next experiment concerns query difficulty. The relevance assessments on the LittleSis dataset show that each query does not have an equal distribution of relevance judgments. In this LittleSis data specifically, most labels are concentrated around the 'Somewhat relevant' and 'Relevant' labels. There are some queries where there are considerably more relevant than non-relevant entities as candidates, making the actual rankings produced by the different methods less important. We are particularly interested in a segment we define as *difficult queries* for this task, estimating difficulty through the proportion of candidates that are judged non-relevant to the ones that are judged relevant in the subgraph. We define *difficult queries* as queries with more non-relevant than relevant candidates.

Table 5.5 shows the performance of the different methods on this particular segment. Interestingly, we observe that the impact propagation method achieves the best performance in this segment. The improvement in terms of P@10 is significant and of a large magnitude (27% improvement). This again confirms the potential of the impact propagation method, since it is successful in retrieving the relevant entities when the subgraph also contains a considerable number of non-relevant entities. This suggests that IP is better at more difficult query entities.

5.4.2 Relation importance analysis

In this section, we answer **RQ3.4**, focusing on the impact propagation method. Recall that one main advantage of our proposed impact propagation method is that it can learn the importance of each relation type in the context of other relations in the subgraph, thus providing us with an interpretable model. This is in contrast with the learning to rank model which learns more generic patterns such as *short paths are more important*

Table 5.4: Contrastive results of LTR vs. IP the LittleSis dataset.

Result	**NDCG@10**	**P@10**
LTR wins	28	18
Ties	13	22
IP wins	13	14

Table 5.5: Results on the LittleSis dataset using the only the difficult query segment, i.e., queries with more judged non-relevant than relevant candidates.

Method	**NDCG@10**	**P@10**
Supervised Random Walks	.709	.552
Learning to Rank	.766	.635
Impact Propagation	**.776**	**.705****

than long paths. One way to interpret this impact propagation model is by looking at the learned weights of each relation directly.

The LittleSis public dataset comes from the political domain, so the model is expected to reflect how impact works in this political universe. Table 5.6 shows the weights of the relations learned by our impact propagation algorithm. We only show the top 10 relations in the table, although there are up to 168 distinct relationships in LittleSis. A quick scan over all relations indicates that some of these relations are less important and would not contribute much to impact-based entity recommendations. The model manages to learn that *campaignDonor-campaignRecipient* in the political domain is important for impact-based recommendation. The model also learns that these campaign-related and transactional relations such as lobbying, contractor, and investor are very important, and weights these key relation types significantly higher than other, more arbitrary *person-company* or *company-to-company* relationships. Similarly, key person-to-organization/company relations such as *foundingPartner* and *institutionalInvestor* are considered more important than more arbitrary relations such as *organization-member*, or social relations such as *close-friends*.

In summary, we conclude that the impact propagation method can learn to distinguish the important relation types, and weight them accordingly. This finding is important because *explainability* provides an added value in our impact-based entity recommendation setting. Although the learning to rank method obtains better performance compared to impact propagation, it can only produce a very generic explanation, while the impact propagation method can estimate the probability of each intermediate nodes and paths leading to the related entities in the subgraph, providing more intuitive explanations.

5.4.3 Error analysis

When looking at specific errors being made by the impact propagation method, we find some interesting cases where the impact propagation performs worse than both the learning to rank and the supervised random walk methods. For one case in particular the

Table 5.6: Relation importance on the public data learned by the impact propagation method. Only the top-10 relations are shown here.

Relation	**Weight**
campaignDonor-campaignRecipient	16.658
campaignRecipient-campaignDonor	15.245
lobbyingClient	3.588
foundingPartner	3.241
directLobbying	2.045
donation	1.887
membership	1.765
contractor	1.622
institutionalInvestor	1.267
client	1.25

connections in the query subgraph are very rich, i.e., there can be up to 33 paths from the query entity to a candidate entity. The query entity has 47 judged candidates which range from somewhat relevant to highly relevant. The relations found in this query subgraph are mostly campaign-related relations, which tend to get very high weights as the model learn that they tend to be important. One possible reason for the impact propagation method to perform worse in this case has something to do with the noisy-OR distribution and the highly connected nature of the subgraph. Recall that the noisy-OR distribution accumulates the impact from multiple paths in a superlinear fashion [174]. Combined with a high degree of connections and highly-weighted relations this would mean that a lot of entities will receive a high probability of impact in this kind of subgraph, which would explain the worse performance of impact propagation compared to other methods.

This finding suggests an investigation of a better causal aggregation method. Possible solutions include adjusting the aggregation by taking into account the number of incoming edges, or even learning how to perform causal aggregation from data directly.

5.4.4 Scalability and efficiency

In this section, we analyze the complexity of the compared methods when learning from M training data and making predictions for C candidates, in particular with respect to the subgraph size S as defined by the number of vertices V and the number of edges E.

Learning The complexity of the supervised random walk method during training is: $O(G(P(V^2F)) + M^2))$ that is G gradient descent iterations with P iterations of the random walk with restart procedure which grows quadratically with respect to the number of nodes V, including computing the derivative for page rank for each of the F feature. Because the algorithm works by optimizing pairwise preferences, the training data available also grows to M^2. While for learning to rank with the Random Forest algorithm: $O(T(V + E)M \log M)$, since we are training T trees with the cost of roughly $V + E$ for feature representation and $M \log M$ for growing each tree. Finally, the impact propagation learning complexity is: $O(GM(V+E))$, because we are performing G gradient descent iterations for M data points, visiting V nodes and E edges for each data point. We can see that both learning to rank and impact propagation are much more efficient during learning, especially with respect to the size of the query subgraphs.

Prediction As to prediction, SRW will take $O(P(V^2))$, taking P iterations until the PageRank converges. For learning to rank, the complexity is $O(CT(V+E))$ if there are C candidates in the subgraph. IP will take $O(C(V+E))$ to make C predictions. All of these algorithms have a worst case complexity of $O(V^2)$ during prediction as E and CV can grow to V^2. However, the impact propagation method can be considered the most efficient one up to a constant factor, while also delivering some improvements over SRW when it comes to effectiveness.

5.5 Conclusions

In this chapter, we have considered *entity-entity associations* and brought the nature of the associations (as specified by the relation type) to the forefront. We have introduced the notion of impact and proposed the novel task of impact-based entity recommendations from knowledge graphs. The novel task can be considered as a form of exploratory search on graph data. The data that we use in our experiments contains a large number of relations, which renders methods that are based on learning combination of relations of path such as [126] ineffective.

To address this task, we propose two novel graph-based recommendation methods: learning to rank and impact propagation. In our learning to rank method, we extract global characteristics of the subgraph connecting query and related entities and learn a model to score the candidate entities. In the impact propagation method, we treat the subgraphs as a Bayesian network and learn shared network parameters of conditional probabilities in a supervised fashion. We have asked the following question:

RQ3 Given graph-based information of entity relations with types, can we effectively recommend related entities based on their direct and indirect connections to a query entity?

To answer RQ3, we have experimented with entity relations stored in a publicly available and an industrial knowledge graph.

Our experiments show that our proposed methods managed to achieve a good performance, showing that the task can be addressed effectively. Our best method outperforms a supervised baseline method based on graph proximity. Our best model achieves 11% performance improvement in terms of precision and 10% improvements in terms of NDCG over this baseline on the industrial dataset. Upon comparing the performance across all queries, we find that the proposed methods outperform each other on different sets of queries. In addition, we also find that the impact propagation method performs better on difficult queries.

Our findings have the following implications. First, our impact propagation method can learn important relation types and comes with explainability; thus, it will be useful to help users who require explanations when exploring the relatedness of entities in a knowledge graph. Second, the recommendation performance is query-dependent; therefore, query features should be incorporated into the recommendation algorithm, as they might affect the size and complexity of the candidate subgraphs.

Our method also has several limitations. First, we are currently only using edge features as parameters in our propagation model. Secondly, our causal aggregation method is limited to the noisy-OR strategy, which performs poorly on some cases. Finally, the explanations that our model provides are still not tailored to general users.

For future work, we are interested in exploring several directions. First, we would like to extend our approach to incorporate query, source, and target node features. Secondly, we would like to experiment with semi-supervised learning, training the model based on pseudo-training data based on known entity associations. Finally, we would like to improve this by generating explanations that are more accessible for the users, as in [234].

6
Document Filtering for Long-tail Entities

In this chapter, we continue with our focus on the second type of association: *entity-document associations*. This type of association can be explored from two directions: the relevance of a document with respect to an entity, or the saliency of an entity within a document. We focus on the former and consider the task of filtering documents for the purpose of updating an entity's knowledge base profile.

A knowledge base contains information about entities, their attributes, and their relationships. Modern search engines rely on knowledge bases for query understanding, question answering, and document enrichment [9, 169, 234]. Knowledge-base construction, either based on web data or on a domain-specific collection of documents, is the cornerstone that supports a large number of downstream tasks. In this chapter, we consider the task of entity-centric document filtering, which was first introduced at the TREC KBA evaluation campaign [77]. Given an entity, the task is to identify documents that are relevant and vital for enhancing a knowledge base entry for the entity given a stream of incoming documents.

To address this task, a series of *entity-dependent* and *entity-independent* approaches have been developed over the years. Entity-dependent approaches use features that rely on the specifics of the entity on which they are trained and thus do not generalize to unseen entities. Such methods include approaches that learn a set of keywords related to each entity and utilize these keywords for query expansion and document scoring [57, 142] as well as text-classification-based approaches that build a classifier with bag-of-word features for each entity [77]. Signals such as Wikipedia page views and query trends have been shown to be effective, since they usually hint at changes happening around an entity [11]; these signals are typically available for popular entities but when working with long-tail entities, challenges akin to the cold-start problem arise. In other words, features extracted from and working for popular entities may simply not be available for long-tail entities.

In this chapter, we are particularly interested in filtering documents for long-tail entities. Such entities have limited or even no external knowledge base profile to begin with. Other extrinsic resources may be sparse or absent too. This makes an entity-dependent document filtering approach a poor fit for long-tail entities. Rather than learning the specifics of each entity, *entity-independent* approaches to document filtering aim to learn the characteristics of documents suitable for updating a knowledge base profile by utilizing signals from the documents, the initial profile of the entity (if present), and relationships between entities

Table 6.1: Glossary of the main notation used in this chapter.

Symbol	**Gloss**
S	Stream of documents
d	a document
e	an entity
p	a profile of an entity
a	an aspect of an entity

and documents [11, 235, 236]. While entity-dependent approaches might be able to capture the distributions of features for each entity better, entity-independent approaches have the distinct advantage of being applicable to unseen entities, i.e., entities not found in the training data. As an aside, entity-independent methods avoid the cost of building a model for each entity which is simply not practical for an actual production-scale knowledge base acceleration system. We ask the following question:

RQ4 How do we filter documents that are relevant to update an entity profile, if the entity is in the long-tail?

Our main hypothesis is that a rich set of *intrinsic* features, based on aspects, relations, and the timeliness of the facts or events mentioned in the documents that are relevant for a given long-tail entity, is beneficial for document filtering for such entities. We consider a rich set of features based on the notion of *informativeness*, *entity-saliency*, and *timeliness*. The intuition is that a document (1) that contains a rich set of facts in a timely manner, and (2) in which the entity is prominent makes a good candidate for enriching a knowledge base profile. To capture informativeness, we rely on three sources: generic Wikipedia section headings, open relations, and schematized relations in the document. To capture entity-saliency, we consider the prominence of an entity with respect to other entities mentioned in the document. To capture timeliness, we consider the time expressions mentioned in a document. We use these features with other basic features to train an entity-independent model for document filtering for long-tail entities.

Our main contributions can be summarized as follows: (1) We propose a competitive entity-independent model for document filtering for long-tail entities with rich feature sets designed to capture informativeness, entity-saliency, and timeliness. (2) We provide an in-depth analysis of document filtering for knowledge base acceleration for long-tail entities.

6.1 Problem Definition

In this chapter, we study the problem of identifying documents that contain vital information to add to a knowledge base. We formalize the task as follows. Given an entity e and a stream of documents S, we have to decide for each document $d_e \in S$ that mentions e whether it is vital for improving a knowledge base profile p_e of entity e. More formally, we have to estimate:

$$P(rel \mid d_e, e), \tag{6.1}$$

where rel is the relevance of document d_e with respect to entity e. A document is considered *vital* if it can enhance the current knowledge base profile of that entity, for instance by mentioning a fact about the entity within a short window of time of the actual emergence of the new fact. Note that a profile p_e is a textual description of an entity (i.e., not a structured object), such as a Wikipedia page or any other web page providing a description of the entity at a certain point in time.

6.2 Method

In this section, we describe our general approach to document filtering. We consider several intrinsic properties of a document that will help to detect vital documents. In particular, we consider the following dimensions:

- **Informativeness** – a document d that is rich in facts is likely to be vital.
- **Entity-saliency** – a document d in which an entity e is salient among the set of entities E occurring in d is likely to be vital.
- **Timeliness** – a document d that contains and discusses a timely event (with respect to document creation time or classification time) is likely to be vital.

We hypothesize that not all of these properties need to be satisfied for a document to be considered vital, i.e., some combination of features derived from these properties and other basic features for document filtering would apply in different cases.

6.2.1 Intrinsic features

Below, we detail the intrinsic features derived to capture the three dimensions described above and how these features are used to operationalize Eq. 6.1. The features are meant to be used in combination with others that are commonly used in document filtering and that will be described below. In the following paragraphs we describe these features; a high-level summary can be found in Table 6.2.

Informativeness features

Informativeness features aim to capture the richness of facts contained in a document. The intuition is that a document that contains a lot of facts, for instance in the form of relations, such as *work-for*, *spouse-of*, *born-in*, is more likely to be vital. We operationalize informativeness in three ways, using entity page sections in a knowledge base (e.g., Wikipedia), open relations, and schematized relations as detailed below. We denote the informativeness features as F_I.

Wikipedia aspects. We define aspects as key pieces of information with respect to an entity. The central idea here is that a vital document contains similar language as some specific sections in Wikipedia pages; cf. [75]. We therefore aggregate text belonging to the same Wikipedia section heading from multiple Wikipedia pages in order to build a classifier. To be able to extract aspect features for a document, we first construct a bag-of-words model of aspects A_c of an entity type c from Wikipedia as detailed in Algorithm 4.

Algorithm 4 Building a Wikipedia aspect model.

```
Input:  Wikipedia entity category: $c$, Wikipedia articles: $W$
Output:  Aspect model: $A_c$;
$C \leftarrow retrieveArticles(W, c)$
$H_C \leftarrow extractSectionHeadings(C)$
$aggregateSectionHeadings(H_C)$
for each $h \in H_C$ do
    $S_C \leftarrow retrieveSections(H_C, h)$
    $a_s \leftarrow combineSections(S_C)$
```

Here we first retrieve Wikipedia articles of *all* entities belonging to the Wikipedia category c; our entities are filtered to be either in the *Person* or *Location* category. Next, we identify the section headings within the articles. We take the m most frequent section headings and, for each section heading, we remove stopwords and aggregate the contents belonging to the same heading by merging all terms that occur in the heading as an aggregated bag-of-words. We then represent each aggregated content section as a bag-of-words representation of aspect $a_k \in A$ and compute the cosine similarity between the candidate document d and aspect a_k to construct an aspect-based feature vector

$$A_k(d) = \cos(d, a_k). \tag{6.2}$$

We refer to the vector A_k as the $ASPECTSIM$ features in Table 6.2.

Open relation extraction. Here, we use the relation phrases available from an open information extraction system, i.e., Reverb [68]. As an open relation extraction system, Reverb does not extract a predefined set of entity relations from text, but detects any relation-like phrases. Given a text as input, it outputs unnormalized relational patterns in the form of triples of an entity, a verb/noun phrase, and another entity. As another feature, we utilize the relational patterns generated by Reverb from the ClueWeb09 corpus [69]. Algorithm 5 details our procedure to generate a list of open relation phrases from this output. Due to the large number of patterns and limited amount of training data, it is not feasible to use all of these patterns as features. Therefore, we select popular phrases out of all available patterns. To this end, we first cluster the relation phrases based on their lemmatized form, obtaining grouped patterns G. Then, we estimate the importance of each pattern group $g \in G$ based on their aggregated count in the ClueWeb09 corpus. That is, we sum the occurrence c_p of each pattern p as the count of group g, obtaining c_g. Finally, we select the n most frequent relation phrases. We compute the feature vector by splitting a document into sentences and, for each relation phrase R compiled in the previous step, we generate a feature vector containing the counts:

$$R_k(d) = count(d, r_k), \tag{6.3}$$

where $count(d, r)$ returns the count of any instances of open relation pattern r in the document d. We refer to the vector R_k as the $RELOPEN$ features in Table 6.2.

Closed relation extraction. The last informativeness feature is based on the occurrence of a set of pre-defined relations within the text of the candidate document. We obtain all

Algorithm 5 Selecting open relation phrase patterns.

```
Input:  Open relation phrases: $P$, Corpus $C$
Output: Ranked open relations model: $R$;
G ← $groupPhrasesByLemma(P)$
for each $g \in G$ do
    for each $p \in g$ do
        $c_p \leftarrow getCount(C, p)$
    $c_g \leftarrow c_g + c_p$
$R \leftarrow selectTopk(G, c)$
```

relation mentions detected in the text by a relation extraction system, the Serif tagger [34]. In our task, the corpus contains annotations of relation types based on the ACE relation schema [58]. We only consider relations involving entities that are a person, organization, or location which amounts to 15 ACE relation types. We construct a vector of the ACE relation types at the document level:

$$S_k(d) = count(d, s_k), \tag{6.4}$$

where $count(d, s)$ is the count of detected relations k in the document. We refer to S_k as the $RELSCHEMA$ features in Table 6.2.

Entity saliency features

The entity saliency features F_E aim to capture how prominently an entity features within a document. Although the basic features (defined in §6.2.2) might capture some notion of saliency, they are focused on the target entity only. We extend this by looking at mentions of other entities within the document. For example, if e is the only entity mentioned in the document then it is probably the main focus of the document.

We define a *full mention* as the complete name used to refer an entity in the document and a *partial mention* as the first or last name of the entity. We introduce the following novel features based on this notion of entity saliency. The first feature is simply the number of entities in the document:

$$DOCENTITIES(d) = |M|, \tag{6.5}$$

where M is the set of all entity mentions. The next feature is the number of entity mentions:

$$DOCMENTIONS(d) = \sum_{e'} n(d, m_{e'}), \tag{6.6}$$

that is, the total number of entity mentions as identified by the Serif tagger. The next feature is the number of sentences containing the target entity e:

$$NUMSENT(d, e) = |S_e|, \tag{6.7}$$

where S_e is the set of all sentences mentioning entity e.

We further define the fraction of full mentions of e with respect to all entity mentions in the document:

$$FULLFRAC(d, e) = \frac{n_{full}(d, m_e)}{\sum_{e'} n(d, m_{e'})}, \tag{6.8}$$

and also include the fraction of partial mentions m_e of e with respect to all entity mentions in the document:

$$MENTIONFRAC(d, e) = \frac{n_{partial}(d, m_e)}{\sum_{e'} n(d, m_{e'})}, \tag{6.9}$$

where $n(d, m)$ counts the number of mentions in document d again obtained by the named entity recognizer.

Timeliness features

Timeliness features F_T capture how timely a piece of information mentioned in the document is. We extract these features by comparing the document metadata containing the document creation time t with the time expressions mentioned in the documents:

$$TMATCH_Y(d) = count(year(t), d), \tag{6.10}$$

where $count(year(t), d)$ counts the occurrences of year expressions of t appearing in the document.

$$TMATCH_{YM}(d) = count(yearmonth(t), d), \tag{6.11}$$

where $count(yearmonth(t), d)$ counts the number of times year and month expressions of t appearing in the document. Finally,

$$TMATCH_{YMD}(d) = count(yearmonthday(t), d), \tag{6.12}$$

where $count(yearmonthday(t), d)$ counts the number of times the year, month, and date expressions of t occur in the document d.

6.2.2 Basic features

This section describes basic features F_B that are commonly implemented in an entity-oriented document filtering system [11, 235].

Document features. Features extracted from document d, capturing the characteristics of d independent of an entity. This includes the length, type, and language of d.

Entity features. Features based on knowledge about entity e including, for instance, the number of related entities in the entity's profile p_e. In addition, we incorporate the length of profile p_e and the type of entity profile available: *Wiki*, *Web*, or *Null*.

Document-entity features. Features extracted from an entity and document pair. This includes the occurrences of full and partial mentions of e in the document as well as the first and last position of occurring. They also include similarity between d and p_e and the number of related entities of e mentioned in the document.

Temporal features. Temporal features extracted from the occurrences of e within the stream corpus S. After aggregating entity mentions in hourly bins, we obtain the counts in the previous k hours before the creation of document d, where $k \in \{1, \ldots, 10\}$.

Table 6.2: Features for document filtering, for an entity e and/or document d.

Feature	**Description**
$SRC(d)$	Document source/type
$LANG(d)$	Document language
$REL(e)$	Number of of related entities of e
$DOCREL(e)$	Number of of related entities of e in d
$NUMFULL(d,e)$	Number of mentions of e in d
$DOCREL(d,e)$	Number of of related entities of e in d
$NUMPARTIAL(d,e)$	Number of partial mentions of e in d
$FPOSFULL(d,e)$	First position of full mention of e in d
$LPOSFULL(d,e)$	Last position of full mention of e in d
$FPOSPART(d,e)$	First position of partial mention of e in d
$LPOSPART(d,e)$	Last position of partial mention of e in d
$SPRPOS(d,e)$	Spread (first position $-$ last position) of mentions of e in d
$SIM_{cos}(d,p_e)$	Text cosine similarity between d and p_e
$SIM_{jac}(d,p_e)$	Text jaccard similarity between d and p_e
$PREMENTION_h(d,e)$	Mention count of entity in the previous h hour before document creation time of d
$DOCLEN_{chunk}(d)$	Length of document in number of chunks
$DOCLEN_{sent}(d)$	Length of document in number of sentences
$ENTITYTYPE(e)$	Type of e (PER, ORG, or FAC)
$PROFILETYPE(e)$	Profile type: *wiki*,*web*, or *null*
$PROFILELEN(e)$	Length of entity profile e
$ASPECTSIM_k(d)$	Cosine similarity between d and $aspect_k$ estimated from Wikipedia
$RELOPEN_k(d)$	Number of normalized open relation phrases k in d
$RELSCHEMA_k(d)$	Number of relation type k in document d
$NUMENTITIES(d)$	Number of unique entity mentions in the documents
$NUMMENTIONS(d)$	Number of entity mentions in the documents
$NUMSENT(d,e)$	Number of sentences in d containing entity e
$FULLFRAC(d,e)$	Number of full mentions of e in the document, normalized by number of entity mentions
$MENTIONFRAC(d,e)$	Number of full or partial mentions of e in the document, normalized by number of entity mentions
$TMATCH_Y(d)$	Number of year expressions of timestamp t in d
$TMATCH_{YM}(d)$	Number of year, month expressions of timestamp t in d
$TMATCH_{YMD}(d)$	Number of year, month, date expressions of timestamp t in d

6.2.3 Machine learning model

Next, we detail our classification-based machine learning model. We formulate the task as binary classification and train a classifier to distinguish vital and non-vital documents using the concatenated vector of all features described previously: $F = F_B \cup F_I \cup F_E \cup F_T$. We train a global model M in an *entity-independent* way, utilizing all training data available for the model. Creating such a general model has the benefit that it can be readily applied to entities that do not exist in the training data.

We use gradient boosted decision trees (GBDT) [79] as our machine learning algorithm. GBDT learns an ensemble of trees with limited complexity in an additive fashion by iteratively learning models that aim to correct the residual error of previous iterations. To obtain the probabilistic output as required by Eq. 6.1, the gradient boosting classifier is trained as a series of weak learners in the form of regression trees. Each regression tree $t \in M$ is trained to minimize mean squared error on the logistic loss:

$$MSE = \frac{1}{n} \sum_{i}^{n} \left(y_i^2 - \left(\frac{1}{1 + e^{pred_i}} \right)^2 \right), \tag{6.13}$$

where y is the training label converted to either 0 or 1 for the negative and positive class, respectively, and $pred$ is the prediction score of the regression tree at data point i. The trees are trained in a residual fashion until convergence. At prediction time, each tree produces a score s_t; these are combined into a final score s, which is then converted into a probability using the logistic function:

$$P = \frac{1}{1 + e^{-s}}. \tag{6.14}$$

We take this output as our estimate of Eq. 6.1. We refer to our proposed entity-independent document filtering method as EIDF.

6.3 Experimental Setup

In this section we detail our experimental setup including the data that we use, the relevance assessments, and the evaluation metrics. Our experiments address the following research questions:

In the beginning of this chapter, we ask the following question:

RQ4 How do we filter documents that are relevant to update an entity profile, if the entity is in the long-tail?

Our experiments are driven by the following research questions derived from RQ4:

RQ4.1 How does our approach, EIDF, perform for vital document filtering of long-tail entities?

RQ4.2 How does EIDF perform when filtering documents for entities not seen in the training data?

RQ4.3 How does EIDF compare to the state-of-the-art for vital document filtering in terms of overall results?

6.3.1 Data and annotations

The TREC KBA StreamCorpus contains 1.2B documents. Roughly half of these (579M) have been annotated with rich NLP annotations using the Serif tagger [78]. This annotated set is the official document set for TREC KBA 2014. Out of these annotated documents, a further selection is made for the Cumulative Citation Recommendation (CCR) task of KBA 2014. This results in the final *kba-2014-en-filtered* subset of 20,494,260 documents, which was filtered using surface form names and slot filling strings for the official query entities for KBA 2014. These documents are heterogeneous and originate from several Web sources: arxiv, classifieds, forums, mainstream news, memetracker, news, reviews, social, and blogs. We perform our experiments on this filtered subset.

The entities used as test topics are selected from a set of people, organizations, and facilities in specific geographical regions (Seattle, Washington, and Vancouver). The test entities consist of 86 people, 16 organizations, and 7 facilities, 74 of which are used for the vital document filtering task. Assessors judged $\sim$30K documents, which included most documents that mention a name from the handcrafted list of surface names of the 109 topic entities. Entities can have an initial profile in the form of *wikipedia*, *web*, or *null*, indicating that no entity profile is given as a description of the entity. In order to have enough training data for each entity, the collection was split based on per-entity cut-off points in time. Some of the provided profile pages are dated after the training time cutoff of an entity. To avoid having access to future information, we filter out entity profiles belonging to those cases. Table 6.3 provides a breakdown of profile types of the test entities.

Table 6.3: Distribution of entity profile types and examples.

Entity profile	**Count**	**Examples**
Wiki	14	*Jeff Mangum, Paul Brandt*
Web	19	*Anne Blair, Bill Templeton*
Null	41	*Ted Sturdevant, Mark Lindquist*

Annotators assessed entity-document pairs using four class labels: *vital*, *useful*, *neutral*, and *garbage*. For a document to be annotated as *vital* means that the document contains (1) information that at the time it entered the stream would motivate an update to the entity's collection of key documents with a new slot value, or (2) timely, new information about the entity's current state, actions, or situation. Documents annotated as *useful* are possibly citable but do not contain timely information about the entity. *Neutral* documents are documents that are informative, but not citable, e.g., tertiary sources of information like Wikipedia pages. *Garbage* documents are documents that are either spam or contain no mention of the entity. The distribution of the labels is detailed in Table 6.4. As our model performs binary classification, we collapse the non-vital labels into one class during training.

One of our proposed features is based on generic Wikipedia sections of *Person* and *Location* entities. For this purpose, we use a Wikipedia dump from January 2012.

Table 6.4: Label distribution in the ground truth.

Label	Training	Test
Vital	1,360	4,665
Useful	5,482	20,370
Neutral	522	2,044
Garbage	3,302	1,961

6.3.2 Experiments

We run three experiments: two main experiments aimed at assessing the performance of EIDF on long-tail entities and on unseen entities, and a side experiment in which we determine the performance on all entities.

Main experiment: Long-tail entities. This main experiment aims to answer **RQ4.1** and adapts the standard TREC KBA setting with one difference: we aggregate the results for different entity popularity segments. We define *long-tail entities* to be entities without a Wikipedia or Web profile in the TREC KBA ground truth data. All training entities are used to train the model and, during evaluation, a confidence score is assigned to every candidate document. All experiments are performed on the already pre-filtered documents using the canonical name of the entities as detailed above. Only documents containing at least a full match of the entity name are therefore considered as input. We focus on distinguishing vital and good documents, and use only documents belonging to these labels as our training data.

Main experiment: Unseen entities. The second main experiment aims to answer **RQ4.2**. We assess the performance of EIDF on *unseen entities*, i.e., entities not found in the training data . We design this experiment as follows. We randomly split the query entities into five parts and divide the training data accordingly. For every iteration we train on the training data consisting only of document-entity pairs of the corresponding entity split and test on the remaining split. We perform this procedure five times, resulting in a 5-fold cross-validation.

Side experiment: All entities. Our side experiment aims to answer **RQ4.3** and follows the standard TREC KBA setting. All entities within the test set are considered in the evaluation (i.e., the results are not segmented) to asses the overall performance of EIDF.

6.3.3 Evaluation

In our experiments, we use the evaluation metrics introduced in the TREC KBA track for the vital filtering task: F_{macro}, and maximum scaled utility (SU). We also compute precision (P), recall (R), and F measure: the average of the harmonic mean of precision and recall over topics. For significance testing of the results, we use the paired t-test.

The main evaluation metric, F_{macro}, is defined as the *maximum* of the harmonic mean of averaged precision and recall computed at every possible threshold θ which separates vital and non-vital documents: $\max(\text{avg}(P), \text{avg}(R))$. The motivation behind this is evaluation setup is as follows. A filtering system will have a single confidence threshold

θ for which the classification performance is maximized. Different systems might have different optimal confidence score calibrations, hence choosing the maximum scores with respect to each system's best threshold would ensure the fairest comparison. Below we explicitly distinguish between F_{macro} and F when reporting our experimental results.

SU is a linear utility measure that assigns credit to the retrieved relevant and non-relevant documents and is computed as follows:

$$SU = \frac{\max(NormU, MinU) - MinU}{1 - MinU},$$

where $MinU$ is a tunable minimum utility (set to -0.5 by default), and $NormU$ is the normalized version of utility function U which assigns two points for every relevant document retrieved and minus one point for every non-relevant document. The normalization is performed by dividing $NormU$ with the maximum utility score (i.e., 2 times the number of relevant documents). The official TREC KBA scorer sweeps over all the possible cutoff points and the reports the maximum SU. To gain additional insight, we also computed SU at the cutoff θ with the best F_{macro}: SU_{θ}.

6.3.4 Baselines

In our main experiments, we consider the following baseline approaches to compare the effectiveness of our approach.

Official Baseline [78]. The official baseline in TREC KBA considers matched name fractions as the confidence score.

BIT-MSRA [235]. A random forest, *entity-independent* classification approach utilizing document, entity, document-entity, and temporal features. This approach achieved the best official performance at the TREC KBA 2013 track.

In our side experiment aimed at assessing the performance of EIDF on all entities we also consider a state-of-the-art entity-dependent approach.

MSR-KMG [114]. A random forest, *entity-dependent* classification approach based on document cluster, temporal, entity title and profession features, with globally aligned confidence score. This approach achieved the best official performance in TREC KBA 2014. We take the team's best automatic run for comparison.

6.3.5 Parameters and settings

Recall that a document filtering system should output an estimate of $P(rel \mid d_e, e)$ (Eq. 6.1). The official KBA setup expects a confidence score in the $[0, 1000]$ range for each decision made regarding a document. To make the initial output of our model compatible with this setup, the probabilities are mapped to a confidence score that falls in this interval by adopting the mapping procedure introduced in [11]—we multiply the probability by 1000 and take the integer value.

Our approach involves two sets of hyperparameters. The first set deals with the machine learning algorithm of our choice. GBDT depends on two key parameters: the number of trees, k, and the maximum depth of each tree, d. The other set of parameters concerns the informativeness features. That is, the number of aspects that we used for the aspects-features, m, and the number of open relation patterns to consider, n.

We perform cross-validation on the training data to select the values of these parameters. For the GDBT parameter we consider $k = [100, 250, 500]$ and tree depth $d = [6, 7]$. For the informativeness parameters, we consider $m = [30, 40, 50]$ for the number of aspects and $n = [150, 200, 250]$ for number of the open relation patterns. We select the combination of parameters which maximize the mean F score across the validation folds, and finally set $k = 100$, $d = 6$, $m = 50$, and $n = 200$.

6.4 Results and Discussion

In this section, we present and analyze our experimental results.

6.4.1 Main experiment: Long-tail entities

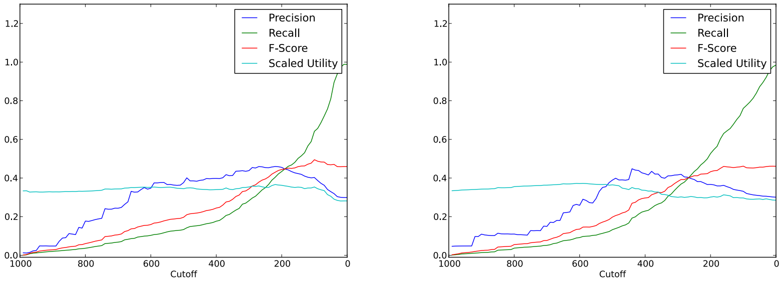
One of our goals in this work is to develop methods that are specifically geared towards filtering documents for long-tail entities. Therefore, we are particularly interested in comparing the performance of the methods on entities with different levels of popularity. To gain insight into our results along this dimension we segment the results by entity popularity using the type of entity profile as a proxy for popularity as defined in §6.3.2. We compute the best threshold for each approach, determine its per-entity performance using this cutoff, and then aggregate the performance by averaging the per-entity scores. We present these results in Table 6.5. Here, we answer **RQ4.1** and compare our approach with other *entity-independent* approaches.

Table 6.5: Results segmented by entity popularity. Significance of EIDF result is tested against the strong baseline (BIT-MSRA). Significant improvement is denoted with ▲ ($p < 0.05$). Here the *null profiles* segment represents the long-tail entities.

Segment	**P**	**R**	**F**	SU_θ
Null profiles				
Official baseline	0.279	**0.973**	0.388	0.268
BIT-MSRA	0.362	0.630	0.404	0.313
EIDF	**0.398**▲	0.645	**0.433**▲	**0.350**▲
Web profiles				
Official baseline	0.391	**1.000**	0.513	0.381
BIT-MSRA	**0.430**	0.867	**0.536**	**0.429**
EIDF	0.424	0.827	0.517	0.410
Wiki profiles				
Official baseline	0.169	**0.975**	0.275	0.044
BIT-MSRA	0.204	0.737	0.296	0.121
EIDF	**0.227**▲	0.704	**0.317**	**0.130**

First, we look at the average scores in each popularity group, starting with the *Null* segment, which represents the long-tail entities in our setting. In the *Null* segment, the recall performance of different methods is considerably lower than on the other two segments, but this is complemented by the fact that precision is higher than for the *Wiki* segment. One important factor in this analysis is that these are most likely tail entities with very few candidate documents to consider. More importantly, our approach achieves a significant improvement in the *Null* segment, while keeping a comparable or better performance as compared to BIT-MSRA on the *Wiki*, and *Web* segments. In particular, the improvements in precision, F, and SU_θ in this segment are statistically significant.

This finding is important because it confirms the effectiveness of our approach in the setting of long-tail entities. Faced with a considerably smaller pool of candidate documents in this segment, EIDF manages to detect more vital documents while simultaneously improving precision. Note that in the TREC KBA 2014 track, long-tail entities constitute a large fraction of the query entities (41 entities, i.e., 56%). The performance of EIDF and BIT-MSRA for long-tail entities across different cutoff points is shown in Figure 6.1.



(a) EIDF

(b) BIT-MSRA

Figure 6.1: Performance of EIDF and BIT-MSRA for long-tail entities across different cutoff points.

Filtering documents for the *Web* profile segment seems to be the easiest relative to the other segments. Recall and precision are highest compared to the other groups, which explains the higher F score. Our approach, EIDF, achieves a P score of 0.424, an F score of 0.517 and SU_θ of 0.410 in this segment. This happens to be lower than the strong baseline (BIT-MSRA), but the differences in performance in this segment are not statistically significant.

Interestingly, the performance of all methods when filtering documents of entities belonging to the *Wiki* group is the lowest. The recall is relatively high, but the F score is brought down by the lower precision. This may be due to the fact that these popular entities have a much larger pool of candidate documents, making the filtering task difficult because a system has to recover only a selective fraction of the documents. Thus, faced with a large set of candidate documents, methods tend to work towards obtaining high recall. Despite this, EIDF manages to get the best precision, obtaining a significant improvement over the strong baseline. The low SU_θ scores indicate that it is difficult to beat a system that returns no relevant documents for this segment group.

After looking at the general performance across the different segments, we compare the performance of our approach against the official TREC KBA baseline. We see considerable gains are obtained in all three segments in terms of precision, F and SU_θ.

Informed by the previous insights, we also perform a follow-up experiment on training segment-conditioned models. Since feature value distributions might be different due to the popularity of an entity, we need to distinguish long-tail entities from more popular ones. One natural way of doing is to consider the existence of a knowledge base profile from Wikipedia—some entities may have a Wikipedia profile, some only an initial profile on a webpage, and some entities have no profile at all. To capture this difference in characteristics, we train three separate machine learning models: M_{wiki} for entities with a Wikipedia page, M_{web} for entities with a lesser profile in the form of a Web page, and M_{null} for entities with no profiles at all. During prediction, the appropriate model is automatically selected and applied to perform the predictions. We failed to obtain any improvements with these segment-conditioned models. This may be due to the fact that by segmenting the data, we lose important information required to train our model with rich feature sets. To fully utilize the data while recognizing the different characteristics of each segment, a learning algorithm that can handle feature interaction, as we employ with tree-based ensembles, seems like a good solution. Having one global model that can handle feature interaction seems to be a better way to handle this problem, without resorting to individual models.

In sum, our approaches achieve the best performance overall across different segments, with the biggest performance gain realized for the long-tail entities segment. Importantly, the features designed for improvement in the long-tail entities segment do not have a significant detrimental effect on the results of other segments. In addition, learning a separate model for each segment does not yield additional benefits.

6.4.2 Main experiment: Unseen entities

In this section, we describe the results of our experiments on answering **RQ4.2**. The results of our experiments with unseen entities are detailed in Table 6.6. Our approach performs best on almost all folds in terms of F_{macro}, gaining significant improvements compared to other approaches on Fold1 and Fold3.

Averaged over all folds, our approach also achieves the best performance. The differences between the performance of different methods in the unseen entities setting is very small in terms F_{macro}. Overall, the learned model tends to be precision-oriented with some loss in recall. Compared to the results of the main experiment (Table 6.5), the result is lower in terms of absolute score. This may be explained as follows. First, the model is now learning on less data—roughly 80% of the full data, depending on the number of data points that contribute to the folds. Secondly, the model is now performing predictions on entities that may have very different characteristics than the ones found in the training data. The average scores in each fold also vary considerably. This can be explained by the fact that by splitting the data in terms of entities, we might end up with different numbers of training and testing data in each split. Additionally, the inherent difficulty of filtering documents within each fold will also vary based on the popularity and the size of the candidate document pools. The magnitude of the improvements obtained in each fold also tends to be smaller, because, with 80% of the data, there are fewer positive examples available to learn a rich set of features (due to the imbalance of *vital*

Table 6.6: Results of cross-validation experiments with unseen entities, in terms of F_{macro} (top), P (middle), and R (bottom).

	Fold1	**Fold2**	**Fold3**	**Fold4**	**Fold5**	**Overall**
Official baseline	0.410	0.482	0.401	0.532	0.400	0.445
BIT-MSRA	0.405	**0.489**	0.413	0.537	0.407	0.450
EIDF	**0.458**	0.485	**0.438**	**0.539**	**0.408**	**0.465**
Official baseline	0.256	0.318	0.252	0.363	0.250	0.288
BIT-MSRA	0.258	**0.324**	0.266	0.371	0.257	0.295
EIDF	**0.328**	0.320	**0.329**	**0.373**	0.257	**0.321**
Official baseline	**1.000**	**1.000**	**0.975**	**0.993**	**1.000**	**0.994**
BIT-MSRA	0.956	0.992	0.923	0.972	0.976	0.964
EIDF	0.762	0.996	0.654	0.973	0.987	0.874

and *non-vital* document labels).

The results of filtering documents for unseen entities are quite promising, and the fact that the learning algorithm is able to achieve a better score than a name fraction baseline indicates that it is successful in learning the characteristics of vital documents and applying it to new, unseen entities.

6.4.3 Side experiment: All entities

To answer **RQ4.3**, we compare our method, EIDF, with *entity-independent* and *entity-dependent* baselines in terms of overall, non-segmented results. Table 6.7 shows the results for this experiment. First, looking at the absolute scores, all methods improve over the official baseline in terms of F_{macro}, SU, and P. The official baseline unsurprisingly achieves the highest recall as it simply considers all document containing exact mentions of the target entity as vital.

Our approach also outperforms the two entity-independent baselines in terms of F_{macro}; we achieve significant improvements over BIT-MSRA in terms of precision, while maintaining the same level of recall. BIT-MSRA achieves a slightly better performance than EIDF in terms of SU. However, the difference is very small and not significant.

Compared to the best entity-dependent approach, EIDF obtains a comparable level of precision and F_{macro}. In summary, EIDF achieves the best entity-independent performance and competitive performance to the state of the art entity-dependent approach.

6.4.4 Feature analysis

Recall that we learn a single, entity-independent model across all entities. We zoom in on the effectiveness of each feature within this global, entity-independent model. The importance of each feature is determined by averaging its importance across the trees that comprise the ensemble model, detailed in Table 6.8.

We observe several things. First, the most important features are a combination of common features in document filtering, e.g., the first position of the entity, the spread

Table 6.7: Overall results with official and additional metrics. Significance of EIDF result is tested against the strong baseline (BIT-MSRA). Significant improvements are denoted with $^\blacktriangle$ ($p < 0.05$). The official TREC KBA scorer returns F_{macro}, SU, P, and R. We also compute additional metrics, F and SU_θ to gain more insight about the results. We can not compute the significance test against MSR-KMG because the run is not available. Due to the way F_{macro} is computed in TREC KBA, as a harmonic mean over recall and precision macro statistics, significance testing cannot be applied to F_{macro}.

Method	**P**	**R**	**F**	SU_θ	F_{macro}	**SU**
Entity-independent						
Official baseline	0.286	**0.980**	0.397	0.253	0.442	0.333
BIT-MSRA	0.348	0.709	0.415	0.305	0.467	0.370
EIDF	0.371$^\blacktriangle$	0.701	0.432$^\blacktriangle$	0.323$^\blacktriangle$	0.486	0.367
Entity-dependent						
MSR-KMG (automatic) [114]	**0.378**	0.744	–	–	**0.501**	**0.377**

Table 6.8: Feature importance analysis for the model learned in the main and side experiments on long-tail entities.

Feature	**Importance**
$FPOSFULL(d,e)$	0.030
$PROFILELEN(e)$	0.025
$FPOSFULL_N(d,e)$	0.022
$REL(e)$	0.021
$ASPECTSIM_{filmography}(d)$	0.019
$DOCLEN_{SENT}(d)$	0.018
$MENTIONFRAC(d,e)$	0.016
$PREMENTION_{h2}(d,e)$	0.016
$SIM_{cos}(d,p_e)$	0.015
$ASPECTSIM_{coachingcareer}(d)$	0.015
$LPOSFULL(d,e)$	0.014
$ASPECTSIM_{politicalcareer}(d)$	0.013
$LSPRFULL_N(d,e)$	0.013
$TMATCH_Y(d)$	0.012
$LPOSFULL_N(d,e)$	0.012
$SIM_{jac}(d,p_e)$	0.012

of entity mentions, and our proposed features. One of our proposed features (profile length) is the most discriminative feature and another of our proposed saliency features, the fraction of entity mentions, is also shown to be quite important. As for the rest, the aspect-based features seem to be the most important features, with as many as three features belonging to the aspect-based group in the top most important features.

The aspect-based features might be complementary to the more common cosine and jaccard profile similarity features. In combination with the profile length feature the aspect-based features seem to be triggered when the profile similarity scores are zero, which will happen in the case of entities without a profile. Having established this, we zoom in on the most important aspect-based features as detailed in Table 6.9. Recall that in our experiments, we use the top-50 aspects constructed from Wikipedia. Often, including aspects-based features seems intuitive, as is the case for, e.g., *achievements*, *accomplishment*, *coaching-career*, and *political-career*, since they are things that are typically included in vital documents.

Table 6.9: Top Wikipedia aspect importance.

Feature	**Importance**
filmography	0.019
coaching-career	0.015
political-career	0.013
wrestling	0.011
references	0.011
championships-accomplishments	0.011
footnotes	0.011
achievements	0.011
selected-publications	0.010
links	0.010

All in all, we extracted 358 features. A breakdown of feature types in the top-30 features is shown in Table 6.10. The informativeness features not ranked among the top in the table are not as discriminative as the Wikipedia aspects. In the case of open relation patterns, some receive a zero relative importance score. One possible explanation is that these patterns are very common and may occur in many documents, thus having very little discriminative power. In other cases, the patterns are quite rare, and they might thus only occur in a few documents.

6.5 Conclusion and Future Work

In this chapter we have addressed an information filtering task for long-tail entities. We have asked the following question:

RQ4 How do we filter documents that are relevant to update an entity profile, if the entity is in the long-tail?

Table 6.10: Feature types within the top-30.

Feature type	Number of features
basic	14
informativeness	13
entity saliency	2
timeliness	1

To answer RQ4, we have developed a method called EIDF. The method incorporates intrinsic features that capture the notions of *informativeness*, *entity saliency*, and *timeliness* of documents. We have also considered the challenges related to filtering long-tail entities and have adjusted our features accordingly. We have applied these features in combination with a set of basic document filtering features from the literature to train an *entity-independent* model that is also able to perform filtering for entities not found in the training data.

Upon segmenting our results by entity popularity, as approximated by its profile type, we have found that our approach is particularly good at improving document filtering performance for long-tail entities. When looking at the overall results of experiments conducted on the TREC KBA 2014 test collection we have found that our approach is able to achieve competitive performance compared to state-of-the-art automatic *entity-dependent* approaches. On filtering documents for unseen entities, we have found that our approach achieves a lower absolute performance overall than on seen entities, as is to be expected, but still improves over a strong name matching and classification baseline. A feature analysis revealed two things. First, entity popularity, proxied using the profile length feature is important. Second, informativeness features, and in particular aspect-based features derived from Wikipedia, are important for this task.

Our results confirm the effectiveness of our entity-independent document filtering approach for knowledge base acceleration for long-tail entities, with (1) its ability to improve filtering performance specifically on the segment of tail entities, and (2) its relatively good performance on classifying documents for unseen entities, i.e., those not found in the training data.

Limitations of the method and analysis that we performed in this work include the following. First, our proposed method does not consider decisions made on previous documents in the stream. Second, we have not analyzed the filtering performance on unseen entities in great detail.

As to future work, we are interested in exploring several directions. First, it would be interesting to explore the effect of combining the proposed features with other machine learning algorithms. Our preliminary experiment in this direction with applying logistic regression as the underlying learning algorithm indicates that we can obtain similar improvements. Next, we aim to apply more semantic approaches such as entity linking to detect entities and concepts mentioned in the context of a target entity. Last, we want to apply incremental learning so as to obtain a document filtering model that is able to learn from its previous decisions.

7

Mining, Ranking, and Recommending Entity Aspects

In this last research chapter, we address the third type of association: *entity-aspect associations*. Entities are often associated with other, more specific information such as the *attributes of the entity*, *topics related to the entity*, and *events involving the entity*. We study these associations in the context of entity-oriented web search, utilizing information obtained from users' search interests in query logs. In the context of web search, understanding these associations might be useful for anticipating information needs in applications such as query recommendation, search result diversification, and knowledge card presentation.

With the proliferation of mobile devices, an increasing amount of available structured data, and the development of advanced search result pages, modern-day web search is increasingly geared towards entity-oriented search [9, 169, 178]. A first step and common strategy to address such information needs is to identify entities within queries, commonly known as *entity linking* [152]. Semantic information that is gleaned from the linked entities (such as entity types, attributes, or related entities) is used in various ways by modern search engines, e.g., for presenting an entity card, showing actionable links, and/or recommending related entities [23, 106, 136].

Entities are not typically searched for on their own, however, but often combined with other entities, types, attributes/properties, relationships, or keywords [178]. Such query completions in the context of an entity are commonly referred to as entity-oriented intents or entity aspects [170, 252]. In this chapter we study the problem of mining and ranking entity aspects in the context of web search. We ask the following question:

RQ5 How can we mine and represent common information needs around entities from user queries? How do we rank and recommend them?

To answer this question, we study four related tasks: (1) identifying entity aspects, (2) estimating the importance of aspects with respect to an entity, (3) ranking entity aspects with respect to a current query and/or user session, and (4) leveraging entity aspects for query recommendation.

The first step in identifying entity aspects involves extracting common queries in the context of an entity and grouping them based on their similarity. We perform this process offline and investigate three matching strategies for clustering queries into entity aspects:

lexical, *semantic*, and *click-based*. Gathering such entity aspects can already be valuable on its own since they can be used to, e.g., discover bursty or consistent entity intents or to determine entity type-specific aspects [170].

In the next step we rank the obtained entity aspects for each entity in a query-independent fashion using three distinct strategies. This provides us with a mechanism to retrieve the most relevant aspects for a given entity on its own, which, in turn, can be used to, e.g., summarize the most pertinent information needs around an entity or to help the presentation of entity-oriented search results such as customized entity cards on SERPs [9].

The third task that we consider is aspect recommendation. Given an entity and a certain aspect as input, recommend related aspects. This task is motivated by the increasing proliferation of entity-oriented interface elements for web search that can be improved by, e.g., (re)ranking particular items on these elements. Recommending aspects for an entity can also help users discover new and serendipitous information with respect to an entity. We consider two approaches to recommend aspects: *semantic* and *behavioral*. In the semantic approach, relatedness is estimated from a semantic representation of aspects. The behavioral approach is based on the "flow" of aspect transitions in actual user sessions, modeled using an adapted version of the query-flow graph [25, 26, 216].

In our final task we leverage entity aspects for actual query recommendation, i.e., helping users refine their query and/or to help users accomplish a complex search task [96, 145]. Most methods for query recommendation are similar to the behavioral approach mentioned above and based on query transitions within sessions. They do not commonly utilize semantic information, however, which may cause distinct but semantically equivalent suggestions. We aim to ameliorate this problem by utilizing the semantic information captured through the entity aspects for query recommendation.

We perform large-scale experiments on both a publicly available and a commercial search engine's query log to evaluate our proposed methods for mining, ranking, and recommending entity aspects, as well as for recommending queries. We perform contrastive experiments using various similarity measures and ranking strategies. We find that entropy-based methods achieve the best performance compared to maximum likelihood and language modeling on the task of entity aspect ranking. Concerning aspect recommendation we find that combining aspect transitions within a session and semantic relatedness give the best performance. Furthermore, we show that the entity aspects can be effectively utilized for query recommendation.

Our main contributions can be summarized as follows:

- We introduce the task of mining, ranking, and recommending entity aspects.
- We provide an in-depth analysis of the mined aspects.
- We propose two approaches to represent aspect relatedness, and utilize them for recommendation.
- We propose a query recommendation method built on top of the query-flow graph.

After a discussion of related work in Section 7.1, we formalize our tasks in Section 7.2. We then detail our approaches to mining, ranking, and recommending entity aspects in Section 7.3. A detailed account of our experiments and data is given in Section 7.4. We discuss the results of our experiments in Section 7.5 and conclude in Section 7.6.

7.1 Related Work

In this section we review related work around three main topics: query intent mining, leveraging entity aspects for search, and search task identification.

Intent mining deals with identifying clusters of synonymous or strongly related queries based on *intents*, which are typically defined as “the need behind a query.” A query intent (sometimes also referred to as an *aspect*) is commonly defined as a set of search queries that together represent a distinct information need relevant to the original search query. Methods for identifying intents are typically based on the query itself, results returned by a retrieval algorithm, clicked results, or any other actions by the user. Hu et al. [109] leverage two kinds of user behavior for identifying query “subtopics” (which can be interpreted as intents): one subtopic per search and subtopic clarification by keyword. They propose a clustering algorithm that can effectively leverage the two phenomena to automatically mine the major subtopics of queries. They represent each subtopic as a cluster containing a number of URLs and keywords. Cheung and Li [49] present an unsupervised method for clustering queries with similar intent and producing patterns consisting of a sequence of semantic concepts or lexical items for each intent. They refer to this step of identifying patterns as *intent summarization*. They then use the discovered patterns to automatically annotate queries.

Other related work focuses on extracting attributes from queries, either unsupervised or in the context of entities from a knowledge base. For instance, Li et al. [132] propose a clustering framework with similarity kernels to identify synonymous query intent “templates” for a set of canonical templates. They integrate signals from multiple sources of information and tune the weights in an unsupervised manner. Li et al. [133] on the other hand, solely aim to discover alternative surface forms of attribute values. They propose a compact clustering framework to jointly identify synonyms for a set of attribute values. In a similar vein, Pasca and Van Durme [171] describe a method for extracting relevant attributes or quantifiable properties for various *classes of objects*. They utilize query logs as a source for these. Yin and Shah [252] propose an approach for building a hierarchical taxonomy of generic search intents for a class of named entities. Their proposed approach finds phrases representing generic intents from user queries and organize these phrases into a tree. They propose three methods for tree building: maximum spanning tree, hierarchical agglomerative clustering, and pachinko allocation model. These approaches are based on search logs only. Moving beyond entity types, Lin et al. [136] introduce the notion of active objects in which entity-bearing queries are paired with actions that can be performed on entities. They pose the problem of finding actions that can be performed on entities as the problem of probabilistic inference in a graphical model that captures how an entity-bearing query is generated.

Another body of related work deals with alternative presentations of search results, e.g., based on intents [50]. For instance, Balasubramanian and Cucerzan [9] propose a method to generate entity-specific topic pages as an alternative to regular search results. Similarly, Song et al. [210] present a model to summarize a query’s results using distinct aspects. For this they propose “composite queries” that are used for providing additional information for the original query and its aspects. This works by comparatively mining the search results of different component queries. Wu et al. [241] mine latent query aspects based on users’ query reformulation behavior and present a system that computes aspects for any new query. Their system combines different sources of information to compute aspects. They first discover candidate aspects for queries by analyzing query logs. They then use a knowledge base to compute aspects for queries that occur less frequently and to group aspects that are semantically related. Finally, Spina et al. [211] explore the task of identifying aspects of an entity given a stream of microblog posts. They compare different IR techniques and opinion target identification methods for automatically identifying aspects.

Search task identification deals with determining the specific task a user is aiming to solve. Such information enables a search engine to, e.g., suggest relevant queries and/or results. Jones and Klinkner [117] formalize the notion of a *search goal* as an atomic information need which results in one or more queries. They propose a method for the automated segmentation of a users' query stream into hierarchical units. While a search goal is atomic, a series of search goals then form a *search missions* (or *complex search tasks*). Lucchese et al. [143] also aim to identify user search tasks within query sessions. They cluster queries in order to find *user tasks*, defined as a set of queries that is aimed towards the same information need. Later they expand user task detection across user sessions [144], similar to so-called task trails and long-term search tasks [134, 235]. Li et al. [129] model the temporal influence of queries within a search session and then use this temporal influence across multiple sessions to identify and label search tasks.

There exists a large body of work on query recommendation, i.e., suggesting follow-up queries to a user, either in an ad hoc fashion or in the context of a user's session or task. Boldi et al. [25] introduces the notion of query-flow graph for query suggestion and Szpektor et al. [216] later expand this model to increase coverage for long tail queries. Bonchi et al. [26] expand it even further to improve coverage. Feild and Allan [72] show that using contextual information can improve query recommendation, as long as the previous queries in the context involve a similar or related task. Hassan Awadallah et al. [96] capitalizes on this idea and propose grouping together similar queries and then using them for query recommendation for complex search tasks, similar to task-specific recommendations [145]. Finally, Verma and Yilmaz [233] extract common tasks in the context of an entity to improve retrieval through query expansion and query term prediction. They extract terms frequently appearing with an entity and aggregate this type of information to an entity type level to obtain a dictionary of entity tasks. They evaluate their work through query term prediction and query expansion.

Our work is different in the following major ways. First, we extract entity aspects from query logs specifically. Second, we weight these aspects and assign their importance with respect to an entity on its own in an ad hoc fashion, i.e., without any user, session or query-based information. Third, we learn their relatedness using semantic and behavioral approaches. Finally, we propose an entity aspect-based query recommendation algorithm building upon the query-flow graph.

7.2 Problem Definition

In this chapter, we study three related entity-oriented tasks that are elemental in modern-day entity-oriented web search: *identifying*, *ranking*, and *recommending* entity aspects. Although they build on one another, we propose effective methods for each of them

Table 7.1: Glossary of the main notation used in this chapter.

t	a term
q	a query
e	an entity
s	a query segment, i.e., a sequence of terms
a	an entity aspect, consisting of zero or more query segments
A_e	the set of entity aspects for e
d	time span

separately since they are essential building blocks for information access applications on their own as well.

In our methods and experiments we employ user interaction log data in the form of queries and clicks. Formally, such logs can be represented as a sequence of events where each event is an action taken by a user. For each event we store a timestamp, a user ID, and the type of action; these are limited to queries and clicks in our current case. Furthermore, the logs are divided into time-ordered sessions, $h \in H$, for each user where we use a common segmentation method and begin a new session after a predefined period of inactivity (30 minutes unless indicated otherwise).

We formulate the first task of *identifying entity aspects* as follows. Given an annotation function $\lambda_e : Q \to E$ that assigns entities from the set of all entities E to queries, we detect "entity-bearing queries," i.e., queries Q_e containing an entity e. Then, for each entity, we mine a set of entity aspects: $A_e = \{a_1, \ldots, a_m\}$ from Q_e representing the key search tasks in the context of that entity. Table 7.1 details the main notation we use in this chapter. We employ the following definition of search tasks and entity aspects.

> *Given a "search task," defined as an atomic information need resulting in one or more queries, an "entity aspect" is an entity-oriented search task, i.e., a set of queries that represent a common task in the context of an entity, grouped together if they have the same intent.*

Once entity aspects have been identified we turn to ranking them. That is, we estimate the importance of each aspect with respect to the entity in a query-independent fashion and rank them accordingly. The obtained ranking can be interpreted as a distribution of prior probabilities over users' information needs on the entity and can be used on its own to, e.g., prioritize an entity display. For entity aspect recommendation, we recommend related aspects in the context of the entity given an entity-bearing query. We also study this problem in a context-aware setting, incorporating previous queries within the search session. Finally, for query recommendation we drop this restriction and include all queries in a session.

7.3 Method

In this section, we introduce our methods for each of the tasks introduced in the previous section.

7.3.1 Identifying entity aspects

To mine aspects for an entity, we first need to identify all queries Q_e that contain entity e from the query log using an annotation function λ_e. Since an entity may be referred to in various ways, we need an effective method for identifying entity mentions in web search queries. Since web search queries are typically short and not grammatically correct [198], we rely on a fairly simple method for entity linking that has been shown to obtain strong performance on such texts [151, 152]. In particular, for each query we generate all possible segmentations and link them to Wikipedia articles. Following [152], we use the CMNS method to generate a link for query segment s:

$$CMNS(e, s) = \frac{|H_{s,e}|}{\sum_{e'} |H_{s,e'}|},$$

where $H_{s,e}$ denotes the set of all links with anchor text s which links to target e in Wikipedia. We start with the longest possible query segments and recurse to smaller n-grams in case no entity mention is detected. In case a segment matches multiple entities, we take the most "common" sense, i.e., the one with the highest CMNS score.

We do not specifically evaluate the performance of our linking method for queries in this chapter. However, in a recent comprehensive comparison on entity linking for queries, CMNS proved to be a very strong unsupervised baseline for this task [24].

Now that we have the set of all queries containing entity e, we remove the mentions of e from the queries and use the remaining segments as *query contexts*. If the query contains more than one entity, we simply consider each entity on its own with the remainder of the query as its context. In this manner, we thus obtain S_e, the set of all context segments which appear with entity e. We then cluster the contexts $s \in S_e$ such that context segments which have the same intent are grouped together. We consider the following features for clustering: *lexical*, *semantic*, and *click* similarity, covering spelling differences/errors, related words/synonyms, and behavioral information. Below we detail the specific methods for each.

Lexical similarity. We compute the lexical similarity between two query contexts using the Jaro-Winkler distance, computed as follows:

$$lex(s_i, s_j) = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_i|} + \frac{m}{|s_j|} + \frac{m-m'}{m}\right) & \text{otherwise,} \end{cases}$$

where m is the number of matching characters, and m' is half the number of transpositions between the two query context segments.

Semantic similarity. To compute the semantic similarity between two query contexts s_i and s_j, we use *word2vec* [156], sum the vectors of each term within the queries, and determine the cosine distance of the resulting vectors:

$$sem(s_i, s_j) = \frac{z(s_i) \cdot z(s_j)}{|z(s_i)| \cdot |z(s_j)|}, \tag{7.1}$$

where $z(s)$ is a function that calculates the semantic vector of s.

Click similarity. Beside lexical and semantic similarity, we also utilize click similarity. In particular, for each query context $s \in S_e$ we obtain all clicked hostnames for all queries containing e and s and combine them into a click vector. We then compute the click similarity between two query segments s_i and s_j using their cosine similarity:

$$click(s_i, s_j) = \frac{c_i \cdot c_j}{|c_i| \cdot |c_j|},$$

where c_i and c_j are click vectors of query s_i and s_j, respectively. The final segment similarity score is calculated by taking the maximum value of the similarity scores.

In order to cluster query contexts into entity aspects, we then employ Hierarchical Agglomerative Clustering (HAC) with complete linkage. We flatten the obtained hierarchical clusters so that for all objects s_i, s_j belonging to the same cluster, $sim(s_i, s_j) \geq \theta$. By the end of this step, we have obtained the entity aspects A_e in the form of the query context clusters.

7.3.2 Ranking entity aspects

The main goal of entity aspect ranking is to estimate the importance of each aspect in the context of an entity in a query-independent fashion. Given an entity and its aspects as input, the output of this task is a list of aspects that is ranked according to their pertinence to the entity. We consider three methods for ranking aspects given an entity. The first model is based on maximum likelihood where we reward more frequently occurring aspects:

$$score_{MLE}(a, e) = \frac{\sum_{s \in a} n(s, e)}{\sum_{a'} \sum_{s \in a'} n(s, e)}, \tag{7.2}$$

Here, $n(s, e)$ denotes the number of times query segment s is queried for in the context of e. Note that this method will not simply place clusters with most members at a higher rank, since there might be clusters with few members in which the members occur more frequently than in a large cluster.

The second model uses entropy-based scoring where we reward the most “stable” aspects using different time granularities including months, weeks, and days. For instance, in the case of days we partition the query log into daily chunks and count the number of times completion s is queried for in the context of e on that day. We then determine the entropy:

$$score_{E_d}(a, e) = \sum_{d \in D} P(a|d, e) \log_2 P(a|d, e), \tag{7.3}$$

where D is the set of all time units and $P(a|d, e)$ the probability of observing any $s \in a$ in the context of e on time interval d (we omit the minus sign in order to make these scores comparable). In another variant we determine the joint entropy, incorporating a factor $p(d|e)$:

$$score_{EJ_d}(a, e) = \sum_{d \in D} P(a, d|e) \log_2 P(a, d|e), \tag{7.4}$$

where $P(a, d|e)$ is the joint probability of observing any $s \in a$ and time interval d in the context of e.

The third and final model is based on language modeling and aims to rank aspects by how likely they are generated by a statistical language model based on a textual representation of the entity. There are differences between the basic MLE method introduced earlier. First, these LM approaches are based on the probablities of terms estimated from unigram language model of entity representation θ_e, while the MLE method are based on estimates at the segment level. More formally, we introduce the following three variants:

$$\begin{aligned} score_{LM}(a, e) &= \prod_{s \in a} P(s|\theta_e) \\ score_{LM\text{-}avg}(a, e) &= \frac{1}{|a|} \sum_{s \in a} P(s|\theta_e) \\ score_{LM\text{-}max}(a, e) &= \max_{s \in a}(P(s|\theta_e)), \end{aligned}$$

where $P(s|\theta_e)$ is determined using maximum likelihood estimation with Dirichlet smoothing:

$$P(s|\theta_e) = \prod_{t \in s} P(t|\theta_e) = \prod_{t \in s} \frac{n(t, r(e)) + \mu P(t)}{\sum_{t'} n(t', r(e)) + \mu}.$$

Here, $r(e)$ is the textual representation of e, for which we consider either the entity's Wikipedia article text, LMW, or the frequency-weighted aggregation of all queries leading to a click on the entity's Wikipedia article, LMC. $P(t)$ is the probability of term t estimated from all textual representations of the type at hand. We set μ to the default value of the average document length.

Note that the main difference with the MLE method introduced earlier lies in the fact that the LM approaches are based on unigram term probabilities whereas the MLE estimates operate at the segment level.

7.3.3 Recommending entity aspects

The goal of this task is to recommend aspects related to the entity and aspect currently being queried. Given such an entity and aspect pair as input, the output of this task is a ranked list of aspects. We consider two methods for recommending entity aspects: a *semantic* and a *behavioral* approach. In the semantic approach, we determine the aspects' relatedness based on semantic similarity. In the behavioral approach, we estimate the relatedness from user sessions, inspired by the query-flow graph [25].

We use a graph-based representation for representing entity aspect relatedness. There are two motivations for this decision. First, having a graph-based representation allows the relatedness of the entity aspects to be computed offline. At retrieval time, we retrieve the candidates and rank them based on previously estimated relatedness. Secondly, having this data encoded as graph allows advanced graph-based compression and recommendation techniques to be applied in the future. We construct a different type of graph for each approach: an *aspect-semantic* graph and an *aspect-flow* graph, respectively.

Semantic approach

For the semantic approach we define the aspect-semantic graph for an entity e as an undirected graph $G_{as} = (V, L, w)$ where:

- The nodes are defined as the set of all aspects for e: $V = A_e$,
- $L \subseteq V \times V$ is the set of undirected edges, and
- $w : L \to (0, 1)$ is a weighting function that assigns a weight w to edges $l_{ij} \in L$.

We construct G_{as} using Algorithm 6 and compute the relatedness between two aspects $a_i, a_j \in V$ using:

$$sem(a_i, a_j) = \frac{z(a_i) \cdot z(a_j)}{|z(a_i)| \cdot |z(a_j)|}, \tag{7.5}$$

similar to (7.1). One main difference with (7.1) is that z is now computed from the mean of the semantic vectors of all query contexts belonging to a. If the relatedness score is above a threshold ϕ, we construct an edge between aspect a_i, and a_j and assign score as weight w_{ij}.

Behavioral approach

The second approach is based on the query-flow graph. Formally, we define an aspect-flow graph as a directed graph $G_{af} = (V, L, w)$ where:

- The set of nodes is $V = A_e \cup \{s, t\}$, i.e., the set of all aspects for e plus additional nodes s and t representing a starting state and a terminal state,
- $L \subseteq V \times V$ is the set of directed edges, and
- $w : L \to (0, 1)$ is a weighting function that assigns a weight w to edges $l_{ij} \in L$.

Here, we estimate the relatedness between query aspects from user sessions. We determine the relatedness from the adjacency of the aspects:

$$adj(H, a_i, a_j) = \sum_{h \in H} countAdjacent(h, a_i, a_j),$$

where $countAdjacent(h, a_i, a_j)$ denotes the frequency of query transitions of any query segment $s \in a_i$ to any segment in a_j found in the user session h, i.e., how often a_j follows a_i. We construct the aspect-flow graph for each entity with Algorithm 7. First, we construct a node for every aspect a that occurs in the user sessions, H. Then, for every pair of aspects (a_i, a_j), we compute their adjacency in H. We create an edge between two aspects a_i and a_j if the adjacency is above a threshold φ. We assign the adjacency count, normalized to transition probability, as the weight w_{ij}.

Generating aspect recommendations

We utilize the aspect-semantic graph G_{as} and aspect-flow graph G_{af} to generate recommendations for an input aspect a in the context of entity e. In the first variant, we generate aspect recommendations without a user session's context as detailed in Algorithm 8. Here,

Algorithm 6 Constructing an aspect-semantic graph for e.

```
Input:  Aspect list: $A_e$
Output:  Aspect-semantic graph: $G$
$G \leftarrow initializeGraph(A_e)$
for each $a_i \in A$ do
    for each $a_j \in A$ do
        $w \leftarrow sem(a_i, a_j)$
        if $w > \phi$ then
            $e \leftarrow createEdge(a_i, a_j, w)$
            $G \leftarrow G \cup e$
```

Algorithm 7 Constructing an aspect-flow graph for e.

```
Input:  Aspect list: $A_e$, User sessions: $H$
Output:  Aspect-flow graph: $G$
$G \leftarrow initializeGraph(A_e)$
for each $a_i \in A$ do
    for each $a_j \in A$ do
        $w \leftarrow adj(H, a_i, a_j)$
        if $w > \varphi$ then
            $e \leftarrow createDirectedEdge(a_i, a_j, w)$
            $G \leftarrow G \cup e$
```

we retrieve candidate recommendations from all nodes adjacent to a in G. For G_{af}, we only retrieve neighboring nodes connected by the outgoing links from a.

We combine the output of both methods to improve the coverage and effectiveness of our recommendations. First, we combine the outputs with a simple round robin strategy, alternating the retrieval of recommendations from the behavioral and the semantic approach, respectively. The intuition is that the semantic method will be able to complement the behavioral method, since it will have higher coverage if constructed with a relatively low threshold ϕ.

We also experiment with another combination method: *convex combination*. We retrieve the scores generated by the behavioral and semantic approaches and combine them with a weight λ:

$$score(a, a') = \lambda \cdot flow(a, a') + (1 - \lambda) \cdot semantic(a, a')$$

Since the scores are on a different scale, we perform *min-max normalization* to each score before combining them. Due to our graph construction process there might be cases where either method can not provide any score; for these we simply assign a zero score.

In a variant of this method, we incorporate context-awareness by looking at the previous queries in a user's search session as detailed in Algorithm 9. First, we retrieve the recommendation candidates from the neighbors of a in G. Then we compute initial recommendation scores for each of them and, lastly, we incorporate scores from any previous aspect a' within the search context S, dampened based on their distance:

$$decay(a, a') = \delta^{|a-a'|},$$

Algorithm 8 Aspect recommendation.

```
Input:  Aspect graph: $G$, Input aspect: $a$;
Output:  Ranked aspects: $R$;
$C \leftarrow getCandidatesFromNeighbors(G, a)$
for each $ca \in C$ do
    $score[ca] \leftarrow getWeight(G, a, ca)$
$R \leftarrow rankCandidates(score)$
```

Algorithm 9 Context-aware aspect recommendation.

```
Input:  Aspect graph: $G$, Input aspect: $a$, Search context: $S$;
Output:  Ranked aspects: $R$;
$C \leftarrow getCandidatesFromNeighbors(G, a)$
for each $ca \in C$ do
    $score[ca] \leftarrow getWeight(G, a, ca)$
for each $p \in S$ do
    $score_p \leftarrow decay(a, p) * getWeight(G, p, ca)$
    $score[ca] \leftarrow score[ca] + score_p$
$R \leftarrow rankCandidates(score)$
```

where δ is a decay constant, and $|a - a'|$ indicates the distance between a and a'. The distance is the number of aspects queried by the user between a and a' in the current session S.

Generating query recommendations

So far, we have focused on problems and approaches for ranking and recommending aspects involving the same entity. In this section, we detail how we leverage entity aspects for query recommendation in general. That is, recommending other entities, other entity aspects, or regular/non-entity queries for a given query. We complement a state-of-the-art query recommendation method—the query-flow graph [25]—with information from the entity aspects.

We first apply the information from entity aspects when constructing the query-flow graph. We preserve all other queries that are not entity queries, thus forming the query nodes as in a regular query-flow graph. For an entity-bearing query q_e, we link all mentions of an entity e. Next, if the query contains additional query context, we extract the context segment s from q_e. Then, we match s to an appropriate aspect a in the aspect model A_e of e. We perform this matching by finding a which contains s as its cluster member. We collapse different mentions of the same entity into one entity node and collapse semantically-equivalent queries into one entity aspect node. This way, we obtain a “semanticized” query-flow graph.

Lastly, we introduce our recommendation method, detailed in Algorithm 10. For every input query, we perform entity linking of the query to detect entity bearing queries (the *annotateQuery* function). Next, we match an entity query to a node (the *matchToNode* function) with the similar procedure applied during graph construction. Regular queries

Algorithm 10 Generating query recommendation (QFG+A).

```
Input:  Input query: $q$, Query graph: G
Output: Ranked recommendations: $R$
for each $q \in Q$ do
    $q* \leftarrow annotateQuery(q)$
    $n_q \leftarrow matchToNode(G, q*)$
    $S \leftarrow getCandidates(G, n)$
    for each $n_{ca} \in S$ do
        $score[n_{ca}] \leftarrow getWeight(G, n_q, n_{ca})$
    $R \leftarrow rankCandidates(score)$
```

will be matched straightforwardly. Lastly, we retrieve recommendation candidates from adjacent nodes, scored by their weights in the graph.

7.4 Experimental Setup

In this section we detail our experimental setup, including the data we use, the relevance assessments,[1] and the metrics we employ.

In the beginning of this chapter we ask:

RQ5 How can we mine and represent common information needs around entities from user queries? How do we rank and recommend them?

Our experiments below address the following research questions derived from RQ5:

RQ5.1 When mining entity aspects, how do different similarity measures compare on the task of clustering queries in the context of an entity?

RQ5.2 How do different aspect ranking methods compare on the task of ranking entity aspects in a query-independent scenario?

RQ5.3 How do the semantic and behavioral approaches compare on the task of aspect recommendation?

RQ5.4 Does incorporating context improve aspect recommendation?

RQ5.5 Can we leverage the semantic information captured through entity aspects to improve the effectiveness of query recommendation built on top of the query-flow graph?

7.4.1 Experiments

To answer our research questions we set up four experiments, which we describe below. In our experiments we test for statistical significance using a paired t-test, indicating significantly better or worse results at the $p < 0.01$ level with ▲ and ▼ respectively.

[1] Our relevance assessments and editorial guidelines are available at `http://ridhorei.github.io/entity-aspects/`.

Evaluating mining entity aspects. In this experiment, aimed at answering **RQ5.1**, we evaluate the quality of the extracted entity aspects by manually evaluating the generated clusters. We use a set of 50 entities sampled from user logs in a stratified fashion. That is, we bias the sample such that more popular entities are more likely to be included. We then extract the query completions for each entity over a period of time from the *dev-contexts* collection (introduced in the next section). To obtain ground truth data we manually cluster the query segments by grouping those that represent the same aspect together. To evaluate the quality of each entity aspect we employ commonly used cluster quality metrics: B-cubed recall (B-recall), precision (B-precision), and F1 (B-F1) [5].

Evaluating ranking entity aspects. The second experiment is aimed at answering **RQ5.2**. Since manually evaluating aspect rankings for entities without any explicit query is not straightforward, we resort to automatic evaluation. We propose an automatic evaluation based on what we call “underspecified” entity queries, that is, queries that contain only an entity. We rely on the assumption that a good aspect ranking is one that, on average, best satisfies users that issue such underspecified entity queries. Specifically, we consider sessions that contain an underspecified entity query and aim to predict any subsequent queries that again contain the entity, plus additional query terms.

For this experiment we consider one month of query logs (the *test-aspect-ranking* collection) that is disjoint from any log data used for training). Because of this disjointness there might be aspects that our method is unable to predict, simply because they have not been seen before. This includes spelling variants, reformulations, and new aspects. In our experiments below we do keep them as relevant samples in the evaluation data in order to mimic a real-life setting as closely as possible.

We consider the following setup: we aim to predict the next query a user issues in a session, only considering pairs of adjacent entity-bearing queries in the session where the second query contains the same entity plus additional query terms. We then observe at which position our method ranks this subsequent query and score it accordingly. Since we only have pairs of queries and thus only one relevant suggestion for each, we report on mean reciprocal rank (MRR) and success rate (SR).

Evaluating recommending entity aspects. The third experiment is aimed at answering **RQ5.3** and **RQ5.4**. Here, we evaluate the effectiveness of recommending entities and aspects in the context of a user session and constituent queries. We follow a similar evaluation approach to the ranking task above, i.e., we consider the next query in the session as the target to predict. As such we again report on mean reciprocal rank (MRR) and success rate (SR).

Since detecting entity-dominated sessions is not trivial, we simulate them through the following procedure. First, we extend the session demarcation boundary, effectively merging the sessions belonging to the same user within a 3-day timeframe (the *test-aspect-recommendation* collection). Then, we consider the first entity within these extended user sessions as the reference entity and evaluate the recommendation methods by their effectiveness in predicting subsequent aspects of the entity throughout the remainder of the session. This setup reflects recommending related entity aspects for complex search tasks in the context of an entity.

Evaluating query recommendation. Our fourth and final experiment addresses query recommendation and is aimed at answering **RQ5.5**. Here we evaluate actual query pair predictions, following the automatic evaluation method from [216]. We sample 1,000,000 query sessions from the query logs of a commercial search engine (the *test-query-recommendation* collection) and extract pairs of adjacent queries from the sessions. Queries belonging to same entity aspect are treated as equivalent queries during evaluation. We evaluate this approach in two configurations: looking at all queries within the sessions (*all-pairs*), and using only the first and last queries (*first-last pairs*). Furthermore, we also differentiate between using all query pair occurrences (allowing possible duplicates of popular queries pairs) and using distinct occurrences only. Our main evaluation metrics for this experiment are again mean reciprocal rank (MRR) and success rate (SR). To gain additional insights, we also look at the fraction of correct predictions at different recommendations cut-off levels: 100 and 10.

7.4.2 Experimental data and settings

We use two sources of data for training and testing, including user logs of the Yahoo web search engine as well as the AOL query logs. From the former we sample a number of datasets. All the development and test datasets that we use are disjoint, i.e., they are sampled from non-overlapping time periods. The *dev-contexts* dataset is a large, 1-year query log sample containing queries that we use to build the full aspect model for our set of entities. The *dev-clicks* dataset is a 1-month sample used to compute click similarity for the context terms. We build our query-flow graph and the aspect-flow graph on the *dev-flow* dataset (a 1-month sample). The test datasets, *test-aspect-ranking*, *test-aspect-recommendation*, *test-query-recommendation*, are all 1-month samples and unseen query logs that are used in our automatic evaluation methods for our second, third, and fourth experiment, respectively. In addition, we also utilize the publicly available AOL dataset in our second experiment. This last dataset includes queries sampled from March 2006 until May 2006. We define navigational queries as queries that are in the top-40% in terms of the number of pageviews and that also lead to a click on the top search result in at least 40% of the cases. We detect and subsequently discard navigational queries based on this heuristic.

We perform entity linking and context term extraction using the method described in Section 7.3.1. Below we focus on Wikipedia entities and we leave using other entity repositories for future work. In order to reduce data sparseness we remove entities that occur in less than 100 queries. This results in a set of about 75k entities of interest.

Our approach involves several parameters. The first parameter, θ, is the similarity threshold used for clustering. From a preliminary experiment on held-out data, we obtain the optimal value of $\theta = 0.75$. For the minimum relatedness score in the construction of the aspect-semantic graph, we set $\phi = 0.1$. Following the common pratice in constructing a query-flow graph [25, 31], we retain only query transitions that appear at least two times, thus $\varphi = 1$. fter a preliminary experiment, we set $\lambda = 0.85$ when combining the semantic and flow scores. The decay parameter is set to $\delta = 0.85$.

7.5 Experimental Results

In this section we answer the research questions presented in the previous section.

Table 7.2: Entity aspect mining results. Significance is tested against the lexical method (row 1).

Method	B-Recall	B-Precision	B-F1
Lexical	0.9164	0.8258	0.8338
Semantic	0.9452$^{\blacktriangle}$	0.7744$^{\blacktriangledown}$	0.8117$^{\blacktriangledown}$
Click	0.8977	0.6666$^{\blacktriangledown}$	0.6880$^{\blacktriangledown}$
Lexical + semantic	0.9216	0.8629$^{\blacktriangle}$	0.8607$^{\blacktriangle}$
Lexical + click	0.8480$^{\blacktriangledown}$	0.8155$^{\blacktriangledown}$	0.7842$^{\blacktriangledown}$
Semantic + click	0.8686$^{\blacktriangledown}$	0.7788$^{\blacktriangledown}$	0.7680$^{\blacktriangledown}$
Lexical + semantic + click	0.8558$^{\blacktriangledown}$	0.8465$^{\blacktriangle}$	0.8098$^{\blacktriangledown}$

Table 7.3: Entity aspect mining: clustering output for entity *Paris Saint-Germain F.C.*.

Cluster	Context terms
1	*real, real madrid vs, vs real madrid, real madrid*
2	*results*
3	*live, live streaming, live stream*
4	*guingamp*
5	*match*
6	*highlights*
7	*transfert*
8	*2013*
9	*monaco, monaco direct, monaco streaming*
10	*om, regarder om*
11	*streaming, en streaming*
12	*barca, barca vs*
13	*barcelona, barcelone*
14	*barcelona vs*
15	*anderlecht*

7.5.1 Mining aspects

Our first experiment concerns mining entity aspects. We start by evaluating the quality of each cluster and then zoom in on the aspects generated during the mining process. Table 7.2 presents the results of using different matching strategies to cluster query context terms into entity aspects. Recall that we perform complete-linkage clustering with a parameter θ as threshold for grouping objects.

First, we look at the results of individual similarity measures. As we can see, using just lexical matching already results in a fairly good B-cubed recall and precision score. This can be explained by the fact that the lexical matching strategy allows minor changes caused by spelling variants or spelling errors, and is successful in performing clustering on these cases. Semantic similarity achieves higher recall at the cost of precision. This means that the clustering method with the current threshold clusters the object aggressively, for

Table 7.4: Entity aspect mining: clustering ground truth for entity *Paris Saint-Germain F.C.*.

Cluster	Context terms
1	*read, real madrid vs, vs real madrid, real madrid*
2	*results*
3	*live, live streaming, live stream, streaming, en streaming*
4	*guingamp*
5	*match*
6	*highlights*
7	*transfert*
8	*2013*
9	*monaco, monaco direct, monaco streaming*
10	*om, regarder om*
11	*barca, barca vs, barcelona, barcelone, barcelona vs*
12	*anderlecht*

example, grouping aspects such as “daughter” and “mother” together. Click similarity has the lowest precision compared to the other two measures for reasons that we will explain below.

Although lexical similarity provides a good start, this strategy fails to group queries that are semantically related. Thus, combining it with semantic similarity improves recall and precision. This combination proves to be the best performing one, compared to using individual measure and all measures.

Adding click similarity with our current strategy does not work well. Upon closer inspection, we find that a lot of unrelated query contexts point to the same host name. For example, contexts related to an entertainer’s news are often directed to the same entertainment site. Therefore, combining clicks with other measures tends to brings down the performance, in particular precision.

Table 7.3 shows sample output generated by our aspect mining method for the entity *Paris Saint-Germain F.C.* (a Parisian soccer club). Our manually created clusters are shown in Table 7.4. The aspect mining produces more clusters than the ground truth. The method fails to group “barca, barca vs, barcelona, barcelone, barcelona vs.”, instead making three clusters from them. With such short strings, the pairwise Jaro-Winkler distance between two objects is bigger than the threshold, thus preventing the objects from being clustered with complete linkage. Also, in some cases the lexical clustering method groups queries that should not be clustered together because they represent different intent/vertical. For instance, two separate clusters (“monaco”) and (“monaco direct, monaco streaming”) should be created instead of putting them together in a single cluster.

Next, we look at the different types of aspect that occur in the context of our example entity *Paris Saint-Germain F.C.*. Many of the aspects refer to a fairly common *transactional* intent for a football club such as “live streaming.” Other sets of intents are *relational*, which concerns the relationship of the topic entity with other entities (in this case, other football clubs). Another set of intents are *categorical*, that is, they deal with type-related intents, e.g., “results”, “match”, “highlights”, and “transfers.” The last aspect we observe concerns attempts to find something related to a certain time point, such as “2013”.

To conclude this section, we formulate our answer to **RQ5.1**. Combining lexical and semantic similarity measures performs best on the task of clustering query context terms for entity aspect mining and we select this method for the remaining experiments. Integrating click similarity tend to hurt performance, particularly precision.

7.5.2 Ranking entity aspects

Our second experiment evaluates the importance of each aspect with respect to the entity in a query-independent setting. The results of this experiment is displayed in Tables 7.5 and 7.6. From these tables we observe consistent results across the two datasets. First, the simple maximum likelihood approach already performs quite well, thus providing a good baseline. The entropy-based methods, in particular using month as time units, achieve the best performance overall, outperforming maximum likelihood and language modeling approaches. We further observe that the absolute scores on the AOL dataset are lower overall which is mainly due to the disjointness nature of the query logs that we use to mine the aspects with the queries in the AOL logs.

We use different granularities of time-slices, and experiment with two variants of entropy-based methods. For the baseline MLE approach, we simply use the aspect popularity over the whole range of our main query log that is used for mining (1 year of data). The different granularities do not really show much difference in terms of performance, although computing entropy on the monthly data provides a slight edge.

There are several possible reasons why language modeling does not work well for this task. First, it is the only approach that does not include any query popularity or frequency information. Secondly, what users search for does not always align with what Wikipedia editors may put in a Wikipedia article—which is in line with previous research [241]. This may result in a so-called knowledge base gap where a user is searching for an important fact that is not included on a Wikipedia article yet. In our case, this may result in low scores with the language modeling approach. Lastly, the fact that the language model based approaches are unigram-based, while the other methods are segment-based might also contribute to the lower scores

We further observe that the absolute scores on the AOL dataset are lower overall which is mainly due to the disjointness nature of the query logs that we use to mine the aspects with the queries in the AOL logs.

We also considered a second experimental variant where we look for entity-bearing query pairs in the whole session. That is, we discard any non-entity bearing queries in between. We find that the scores using this variant are comparable to those reported here.

It is important to note that we compare a different and diverse set of features, with appropriate functions defined for each type of features. However, since the previous step (clustering query contexts into aspects) are kept constant across method, we argue that this comparison is still valuable despite having to compare the combination of features and functions simultaneously. The end-to-end aspect ranking scores should be comparable.

To answer **RQ5.2**, the results show that ranking entity aspects can be done successfully, resulting in sensible absolute MRR and SR scores and we find that entropy-based methods are the best in ranking entity aspects in a query-independent scenario.

Table 7.5: Entity aspect ranking: results on *test-aspect-ranking*. Significance is tested against the MLE baseline (row 1).

	MRR	SR
MLE	0.1931	0.1110
E_{days}	0.2013▲	0.1149▲
E_{weeks}	0.2015▲	0.1139▲
E_{months}	0.2031▲	0.1162▲
EJ_{days}	0.2020▲	0.1208▲
EJ_{weeks}	0.2048▲	0.1259▲
EJ_{months}	0.2052▲	0.1262▲
LMW	0.0431▼	0.0135▼
LMW_{avg}	0.0657▼	0.0200▼
LMW_{max}	0.0583▼	0.0170▼
LMC	0.0488▼	0.0176▼
LMC_{avg}	0.0859▼	0.0313▼
LMC_{max}	0.0755▼	0.0259▼

Table 7.6: Entity aspect ranking: results on the AOL dataset. Significance is tested against the MLE baseline (row 1).

	MRR	SR
MLE	0.0647	0.0340
E_{days}	0.0710▲	0.0383▲
E_{weeks}	0.0712▲	0.0376▲
E_{months}	0.0709▲	0.0376▲
EJ_{days}	0.0684▲	0.0383▲
EJ_{weeks}	0.0692▲	0.0394▲
EJ_{month}	0.0692▲	0.0395▲
LMW	0.0328▼	0.0132▼
LMW_{avg}	0.0457▼	0.0190▼
LMW_{max}	0.0426▼	0.0170▼
LMC	0.0341▼	0.0146▼
LMC_{avg}	0.0499▼	0.0219▼
LMC_{max}	0.0466▼	0.0195▼

Table 7.7: Aspect recommendation: results. Significance is tested against row 1.

Method	MRR	SR
Aspect-semantic	0.0431	0.0244
Aspect-flow	0.0602▲	0.0451▲
Aspect-combined-rr	0.0674▲	0.0486▲
Aspect-combined-convex ($\lambda = 0.85$)	0.0650▲	0.0465▲

Table 7.8: Aspect recommendation: results of context aware experiment where m denotes the context size, i.e., the number of queries used as context. Significance is tested against row 1 of each group.

Method	MRR	SR
Aspect-semantic	0.0431	0.0244
CA-aspect-semantic ($m = 3$)	0.0436▲	0.0248▲
CA-aspect-semantic ($m = 10$)	0.0436▲	0.0248▲
Aspect-flow	0.0602	0.0451
CA-aspect-flow ($m = 3$)	0.0583▼	0.0438▼
CA-aspect-flow ($m = 10$)	0.0583▼	0.0438▼

7.5.3 Recommending aspects

Our third experiment evaluates the quality of recommending entity aspects within a session, comparing the semantic and behavioral approaches (**RQ5.3**). The results are shown in Tables 7.7 and 7.8. Overall, we see that the behavioral aspect-flow approach outperforms the purely semantic approach. When combined, they give the best performance. In our experiments, we experiment with a round-robin and a convex approach to combine the results.

Upon looking at the graphs created by both methods, we see that the semantic method tends to generate larger graphs, thus attaining more coverage. This is not surprising since the flow graph is only constructed with a single month of data. Despite the sparsity and lack of coverage, the behavioral flow-based approach still manages to outperform the semantic approach, providing better quality recommendations overall. Both combination approaches succesfully improve over the individual approaches.

As for incorporating user session context (**RQ5.4**), we observe that the semantic approach gains a small improvement by incorporating previous queries in the search context. However, the flow-based approach performs slightly worse when previous query is taken into account. This is related to the sparsity/lack of transition data on the flow-based approach. The size of the context have little effect in the current adjacency-based recommendation setup.

In conclusion, we find that the behavioral approach is better than the semantic approach for recommending entity aspects, and they can be combined to generate better recommendations.

7.5.4 Recommending queries

In this section we investigate whether our aspect model can help to complement the query-flow graph for query recommendation (**RQ5.5**). Table 7.9 shows the result of predicting all queries within the sampled user sessions (*all-pairs*). Table 7.10 shows the results of using the first query as input to predict only the last query of a session (which can be considered as yielding the desired results).

In the *all-pairs* prediction setup, we see that the aspect-based query recommendation method (labeled *QFG+A* in the table) successfully improves upon the baseline query-flow graph in terms of predictions coverage and ranking. The improvement is small, but consistent and significant across the two different configurations of our experiment. Overall, we achieve around 1% improvement on the correct prediction's coverage. For ranking, our method achieves slightly better mean reciprocal rank and average position of the target query (averaged for correct predictions at the top-100). Considering the large number of queries, these improvements are quite substantial. The improvements become more pronounced as we look at the unique query occurrences rather than all occurrences.

We notice consistent results in the *first-last* experiment also. Improvements in terms of prediction coverage are around 1%, while the ranking also shows consistent improvements.

7.6 Conclusion

In this chapter we have explored *entity-aspect associations* in the context of entity-oriented web search. We have asked the following question:

RQ5 How can we mine and represent common information needs around entities from user queries? How do we rank and recommend them?

To answer **RQ5**, we have developed and evaluated methods for mining entity aspects, ranking their importance, and recommending them directly or leveraging them for query recommendation. We have done so by linking entities within queries, extracting the queries' context terms, and clustering them together into entity aspects if they refer to the same intent.

We have performed four sets of experiments. For mining entity aspects, we found that combining the *lexical* and *semantic* matching strategies performs best for this task. We ranked the obtained entity aspects for each entity using three strategies: *maximum likelihood*, *entropy*, and *language modeling*. We found that the entropy-based methods yield the best performance. For aspect recommendation, we considered two approaches: *semantic* and *behavioral* and found that the latter provides superior results. In our final task we leveraged entity aspects for actual query recommendation. We found that resolving entities and grouping queries into aspects helps to improve query recommendation in a semantic way.

Our results have the following implications. First, search diversification algorithm and knowledge card designers should incorporate the importance of aspects from query logs. Second, query recommendation methods should consider semantic information from entities and aspects.

There are also some limitations to our work. First, our entity linking method only links one entity for each query. Second, we have not experimented with combining signals for aspect ranking. Third, when using semantic features, we simply averaged the vector of each word within the query context terms; more sophisticated embedding method can be incorporated.

As to future work, we would like to extend the study in following directions. First, we would like to incorporate more features and study the performance of aspect mining in a large-scale setting. Secondly, we would like to the see whether the different aspect ranking methods performs differently for different entity types, or different query triggers (user input, auto-completion, related search, etc). Lastly, we would like to incorporate more advanced recommendation methods for query recommendation, e.g., methods based on personalized PageRank.

Table 7.9: Query recommendation: results on the *all-pairs* dataset for each configuration.

	all occurrences	
	QFG	QFG+A
% pairs in all	0.111	0.123
% pairs in top-100	0.076	0.084
% pairs in top-10	0.042	0.047
% pairs in top-1 (SR)	0.015	0.016
MRR	0.024	0.026
avg. position	20.38	20.09
	unique occurrences	
	QFG	QFG+A
% pairs in all	0.108	0.130
% pairs in top-100	0.070	0.088
% pairs in top-10	0.039	0.048
% pairs in top-1 (SR)	0.013	0.015
MRR	0.021	0.026
avg. position	20.57	20.01

Table 7.10: Query recommendation: results on the *first-last* dataset for each configuration.

	all occurrences	
	QFG	QFG+A
% pairs in all	0.147	0.165
% pairs in top-100	0.097	0.110
% pairs in top-10	0.055	0.062
% pairs in top-1 (SR)	0.019	0.021
MRR	0.031	0.034
avg.position	19.20	18.98
	unique occurrences	
	QFG	QFG+A
% pairs in all	0.104	0.126
% pairs in top-100	0.062	0.079
% pairs in top-10	0.031	0.041
% pairs in top-1 (SR)	0.009	0.011
MRR	0.016	0.021
avg.position	21.84	20.90

8 Conclusions

In this thesis, we have explored the broad problem of computing entity associations for search. Our exploration focused on three types of association: *entity-entity*, *entity-document*, and *entity-aspect*. We have considered several algorithmic tasks stemming from these types of association and developed methods in various domains.

Beginning with entity-entity associations in Chapter 3, we considered the problem of ranking related entities from a text document only. In Chapter 4, we considered an important attribute of entity relations: the temporal boundary. We proposed a method to perform classification of temporal evidence and investigated its impact on end-to-end temporal relation extraction performance. In Chapter 5 we considered the task of recommending related entities given direct and indirect entity connections in a knowledge graph. In Chapter 6 we continued with the second theme: the association between entities and documents and considered the task of document filtering for entities. We focused on the long-tail nature of some query entities and developed a method to improve the filtering performance for such entities. Finally, in Chapter 7, we considered aspects of information related to an entity, exploring them in the context of Web search. We introduced the tasks of mining, ranking and recommending them and showed how these aspects could be leveraged for search.

Below, we provide a detailed summary of the contributions and results obtained in the thesis. We answer the research questions introduced at the beginning of the thesis and conclude with future research directions.

8.1 Main Findings

Entity-entity associations

Starting with the *entity-entity associations* theme, we turned to our first study on entity associations to support exploration of a document collection. As only text information is assumed to be available, the method introduced to rank related entities uses text only. In Chapter 3, we asked the following question:

RQ1 How do we rank related entities to support the exploration of a document collection relying on signals from the text alone?

We refined RQ1 into the following specific questions:

RQ1.1 How do related entity ranking methods based on association measures and relation extraction compare?

RQ1.2 Can we combine these various scoring methods in an ensemble to improve the performance?

RQ1.3 How does performance differ across different queries?

We formalized the task of entity network extraction as ranking related entities, in which a query entity is used as the input against which candidate entities are ranked. To answer the above questions, we proposed a learning to rank approach for related entity ranking. We proposed a successful approach that combines co-occurrence statistics from the source context and a classification-based approach inspired by work in relation extraction. We combined features from different association measures and the output of relation extraction classifiers.

Our learning to rank model based on such features can be used for entity ranking in this text-only setting. In addition, we find that the performance of different methods is *query-dependent*. We found that some query entities are more difficult than others due to the overall style and quality of the snippets in which the query entities are mentioned. Some snippets contain direct/indirect speech, or contain invalid co-occurrences due to text preprocessing errors; entities from such snippets are ill-suited to be included as related entities.

In the next chapter, we shifted our attention towards fluent relation types. Specifically, we turned to establishing temporal boundaries of entity relations and focused on the temporal evidence classification task. We asked the following question:

RQ2 How can we effectively classify temporal evidence of entity relations?

We expanded RQ2 into the following questions:

RQ2.1 How does a purely supervised approach with different features and learning algorithms perform on the task of temporal evidence classification?

RQ2.2 How does the performance of a distant supervision approach compare to that of a supervised learning approach on the task of temporal evidence classification?

RQ2.3 How does the performance of a prior-informed distant supervision approach compare to that of a basic distant supervision approach on the task of temporal evidence classification?

RQ2.4 How do the approaches listed above compare in terms of their performance on the end-to-end temporal relation extraction task?

To answer these questions, we isolated the task of temporal evidence classification from the broader task of temporal relation extraction. We focused on this subtask so as to gain more insights specifically on the evidence classification part where most performance gain can be achieved. We developed an approach based on the distant supervision paradigm and set to address the distribution mismatch of the distant supervision (i.e., source) corpus and target corpus. We employed a sampling method for distant supervision, in which distant supervision samples are generated, and then training examples are further re-sampled from the initial pool of automatically generated examples according to an empirical distribution.

We started with a preliminary experiment using a supervised approach and investigated the impact of using different classifiers and text representations. We found that Support Vector Machines and Naive Bayes combined with a complex text representation work best for the purely supervised setting. When considering the distant supervision approach, we found that when the distant supervision was applied as-is, its performance was worse than the supervised approach. We showed that the sampling strategy we proposed successfully improved the performance of subsequent models trained on the corrected training data from a distant supervision procedure. When examining the end-to-end temporal relation extraction performance, we found that the distant supervision approach consistently achieves improvements over the supervised approach.

Moving away from the initial scenario where entities are ranked based on co-occurrence in text only, we turned to a scenario where the underlying relationships between entities are known. Specifically, we continued with the task of related entity recommendation given entity relationships from knowledge graphs in Chapter 4. We introduced the notion of *impact* in a knowledge graph and asked the following question:

RQ3 Given graph-based information of entity relations with types, can we effectively recommend related entities based on their direct and indirect connections to a query entity?

We expanded RQ3 into the following questions:

RQ3.1 How do our proposed methods and the baseline perform on the task of impact-based entity recommendations?

RQ3.2 How does the impact propagations method compare against the learning to rank method?

RQ3.3 How do our proposed methods perform across different queries?

RQ3.4 Can the impact propagation model learn which relationship types are important for impact-based recommendation?

To answer these questions, we experimented with three methods on highly-heterogeneous knowledge graphs from two domains. Our first method was based on learning to rank from subgraph features. The second method that we proposed, impact propagation, is inspired by treating knowledge graph connections similar to dependencies in a Bayesian network. As a baseline, we modified a random walk method for graph proximity computation. We collect judgments from the two knowledge graphs through crowdsourcing, asking annotators to asses the notion of *impact* from entity connections.

We showed that we could effectively rank entities for impact-based recommendations. Moreover, both approaches that we proposed successfully improved upon a baseline based on graph proximity. The learning to rank method achieved slightly better performance than the impact propagation method. However, when we zoomed in on the performance across queries, we found that not all queries were of equal difficulty. The difference in the performance can be attributed to the complexity of the subgraph traversed from the query entity, whether the size of subgraphs or heterogeneity of the relations contained in the subgraph. When we focused on the difficult queries only, i.e., queries where the are an equal proportion of related and non-related entities, we found that the impact propagation performed best. In addition, we found that impact propagation can learn relation importance properly, resulting in an intuitive model that has the advantage of explainable local prediction of directly connected entities, which in turn allows end-to-end recommendation explanation.

Entity-document associations

We moved to the theme of entity-document associations and considered the task of document filtering for knowledge base acceleration in Chapter 6. We focused on the relevance of a document given a query entity in the context of filtering documents that are useful for updating a query entity's knowledge base profile. We focused on long-tail entities and asked the following question:

RQ4 How do we filter documents that are relevant to update an entity profile, if the entity is in the long-tail?

We proposed a method called EIDF, which is tailored towards long-tail entities, and refined RQ4 into the following questions:

RQ4.1 How does our approach, EIDF, perform for vital document filtering of long-tail entities?

RQ4.2 How does EIDF perform when filtering documents for entities not seen in the training data?

RQ4.3 How does EIDF compare to the state-of-the-art for vital document filtering in terms of overall results?

To answer these questions, we experimented with the TREC Knowledge Base Acceleration setup and focused on entity-independent models for document filtering. We introduced features to represent document based on three key intuitions: *informativeness*, *entity-saliency*, and *timeliness*. We instantiated these key notions into intuitive features for document representation and combined them with other features commonly used for filtering.

We showed that our proposed approach improves the performance of vital document filtering on the segment of long-tail entities; the improvements obtained do not sacrifice the overall performance of the method on other popularity segments. As to filtering documents of entities not found in the training data, we obtained improvements of a smaller magnitude, which we suspect was due to having fewer training examples available in the cross-validation experiments.

Entity-aspect associations

As the last theme, we explored the association between entities and common information needs attached to them. Focusing on the Web search domain, we introduced the notion of entity aspect, defined in this domain as *common search tasks in the context of an entity*. We mined query logs of a commercial search engine to obtain aspects in Chapter 7, and asked following question:

RQ5 How can we mine and represent common information needs around entities from user queries? How do we rank and recommend them?

We detailed RQ5 into the following questions:

RQ5.1 When mining entity aspects, how do different similarity measures compare on the task of clustering queries in the context of an entity?

RQ5.2 How do different aspect ranking methods compare on the task of ranking entity aspects in a query-independent scenario?

RQ5.3 How do the semantic and behavioral approaches compare on the task of aspect recommendation?

RQ5.4 Does incorporating context improve aspect recommendation?

RQ5.5 Can we leverage the semantic information captured through entity aspects to improve the effectiveness of query recommendation built on top of the query-flow graph?

To answer these questions, we formalized the tasks of mining, ranking and recommending entity aspects. For mining entity aspects, we focused on evaluating the performance of clustering different query context terms representing the same aspects and created clustering ground truth. On the remaining tasks, we designed automatic evaluation methods utilizing a commercial search engine's query logs.

We found that a combination of lexical and semantic similarity works best when grouping query context terms to be used as aspects. When comparing different methods for estimating the importance of an aspect with respect to the query entity, we found that entropy-based methods, which reward stable information needs, were the most suitable. When we considered the obtained aspects for recommendations, we found that behavioral similarity is the most effective, but semantic similarity was still useful when behavioral signals are sparse. We did not see any improvements when incorporating context for aspect recommendation. When we tried to incorporate entity aspects for query recommendation, we found that semanticizing queries through entities and aspects provided small, but significant improvements.

8.2 Future Work

This thesis resulted in new task formalizations, algorithms, and insights on computing entity associations for search. Beyond the tasks that we have explored in this thesis, interesting new problems and applications emerge. Below, we briefly discuss a selection of possible future work.

Incorporating attributes of associations. As we have explained in Chapter 4, entity-entity associations can be enriched with additional attributes, such as temporal boundary and magnitude (i.e., the strength of the association). In Chapter 3, the association strengths are computed from entity co-occurrences in text, while in Chapter 5 the magnitudes are obtained from external sources. Such temporal and magnitude attributes would naturally be important for any tasks that utilize entity-entity associations; however, incorporating them into ranking or recommendation methods is a challenge.

In the methods that we presented in Chapter 5, we have briefly touched upon this challenge by including them as features. We demonstrated a way to do so with one relation attribute, i.e., magnitude. This strategy is suitable for scalar attributes such as magnitude; for other types of attributes, further investigations are required. For example, temporal attributes will need to be transformed into scalar values, a transformation that is non-trivial.

Online models. In some of the tasks that we considered, e.g., document filtering in Chapter 6, the notion of temporality and entity profile are important. An entity profile tends to be dynamic, as it needs to be updated with specific changes happening to the entity. In this document filtering setting, we should consider a model that can be updated once it makes a decision on a document. In the method that we proposed in Chapter 6, we did not explicitly perform a comparison of the new document to any updated profile. One could argue that keeping track of selected documents and comparing every new document to this portfolio of entity profile documents are the way go. However, this is not always practical given the relatively large document stream in the scenario. In most scenarios, we do not know which information is being updated by a newly discovered document. Simply combining document text is not a feasible solution.

An ideal method should be able to pick up the specific information that the newly discovered document contains, and only add that highlighted part to the entity profile. The challenge lies in finding a way for comparing entity representations and the candidate documents effectively and efficiently. We need to represent the specific piece of information triggering the profile update so as to allow easy comparison and updating. Relation extraction approaches can be one way to address this problem, however not all noteworthy events can be conveniently represented through relation schemas.

Aspect-based knowledge graphs. In Chapter 7, we moved beyond a common knowledge graphs paradigm towards anticipating information needs based on users' search queries. We have considered the task of recommending entity aspects, explored in the context of aspects belonging to an entity. A more interesting and challenging scenario for this task is recommending entity aspects for which the suggested aspects can belong to different entities. This way, we incorporate the fact that users' interests might shift towards other entities (and other entities' aspects) during exploration.

Taking this idea one step further would be to combine a common knowledge graph schema with a user-driven schema mined from entity aspects. Integrating such a user-driven schema is not trivial; some aspects are entity-specific, while others might be shared across entities or entity types. The challenge lies in normalizing such shared aspects so that they can be incorporated efficiently.

Explainable predictions. A lot of the tasks that we proposed in this thesis, e.g., in Chapter 3, 5, 6 were explored in the context of supporting end-users in performing research, analysis, and decision making. This will have broader implications in the sense that, for any of the entity-related tasks that we aim to solve, it is important to develop models that are intuitive or extensible with explanations. In a lot of cases, a simple prediction output such as a scalar value or classification label is simply not satisfactory anymore. Thus, building a predictive model with explainable prediction becomes an important requirement.

This requirement can be solved by designing intuitive models. The impact propagation approach to entity recommendation that we proposed in Chapter 5 is one such model as the final prediction can be explained by a series of intermediate predictions. Although this should already be valuable in principle, end-users without sophisticated technical background would prefer intuitive explanations (e.g., in the form of short text/narration). With this intuitive model, one strategy that can be pursued is text generation: producing text based on the structure and the prediction of the model. However, some models have complex, non-intuitive structures, such as neural networks [128, 193]. Another alternative to pursue in this case is to add the explanation later by learning a separate explanation model to explain the output of the more complex predictive model [193].

Bibliography

[1] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, DL '00, 2000. (Cited on pages 27 and 54.)

[2] E. Alfonseca, K. Filippova, J.-Y. Delort, and G. Garrido. Pattern learning for relation extraction with a hierarchical topic model. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, ACL, 2012. (Cited on page 27.)

[3] A. Alhelbawy and R. Gaizauskas. Graph ranking for collective named entity disambiguation. *ACL*, pages 75–80, 2014. (Cited on page 36.)

[4] J. Allan. Introduction to topic detection and tracking. In *Topic Detection and Tracking*, pages 1–16. Kluwer Academic Publishers, 2002. (Cited on pages 4 and 24.)

[5] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486, 2009. (Cited on page 127.)

[6] M. Asahara and Y. Matsumoto. Japanese named entity extraction with redundant morphological analysis. In *HLT-NAACL*, 2003. (Cited on page 15.)

[7] N. Bach and S. Badaskar. A survey on relation extraction. *Carnegie Mellon University*, 2007. (Cited on pages 18 and 26.)

[8] L. Backstrom and J. Leskovec. Supervised random walks: Predicting and recommending links in social networks. In *WSDM '11*, 2011. (Cited on pages 80 and 89.)

[9] N. Balasubramanian and S. Cucerzan. Topic pages: An alternative to the ten blue links. In *IEEE-ICSC 2010*, 2010. (Cited on pages 97, 115, 116, and 117.)

[10] K. Balog and R. Neumayer. A test collection for entity search in DBpedia. *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 737, 2013. (Cited on page 42.)

[11] K. Balog and H. Ramampiaro. Cumulative citation recommendation: Citation vs. Ranking. *Proceedings of the 36th international ACM SIGIR conference on Research and Development in Information Retrieval*, 2013. (Cited on pages 4, 25, 97, 98, 102, and 107.)

[12] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, 2006. (Cited on pages 40 and 41.)

[13] K. Balog, P. Serdyukov, A. P. D. Vries, P. Thomas, and T. Westerveld. Overview of the TREC 2009 entity track. *TREC*, 2009. (Cited on pages 40 and 44.)

[14] K. Balog, P. Serdyukov, and A. P. de Vries. Overview of the TREC 2011 entity track. In *TREC 2011 Working Notes*. NIST, 2011. (Cited on page 52.)

[15] K. Balog, Y. Fang, M. de Rijke, P. Serdyukov, and L. Si. Expertise retrieval. *Foundations and Trends in Information Retrieval*, 6(2-3):127–256, 2012. (Cited on page 55.)

[16] K. Balog, H. Ramampiaro, N. Takhirov, and K. Nørvåg. Multi-step classification approaches to cumulative citation recommendation. In *OAIR '13*, pages 121–128. Le Centre De Hautes Etudes Internationales D'Informatique Documentaire, 2013. (Cited on page 25.)

[17] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, 2007. (Cited on pages 27 and 54.)

[18] J. Bao, N. Duan, M. Zhou, and T. Zhao. Knowledge-based question answering as machine translation. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 967–976, 2014. (Cited on page 1.)

[19] H. Bast, B. Buchhold, and E. Haussmann. Relevance scores for triples from type-like relations. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, 2015. (Cited on page 22.)

[20] B. Bi, H. Ma, B.-J. P. Hsu, W. Chu, K. Wang, and J. Cho. Learning to recommend related entities to search users. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 139–148. ACM, 2015. (Cited on pages 2, 45, and 79.)

[21] D. M. Bikel, R. Schwartz, and R. M. Weischedel. An algorithm that learns what's in a name. *Machine Learning*, 1999. (Cited on page 15.)

[22] L. Bing, W. Lam, and T.-L. Wong. Wikipedia entity expansion and attribute extraction from the web using semi-supervised learning. *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, pages 567–576, 2013. (Cited on page 20.)

[23] R. Blanco, B. B. Cambazoglu, P. Mika, and N. Torzec. Entity recommendations in web search. In *ISWC '13*, 2013. (Cited on pages 2, 4, 45, 47, 79, and 115.)

[24] R. Blanco, G. Ottaviano, and E. Meij. Fast and space-efficient entity linking for queries. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, 2015. (Cited on pages 1, 37, 79, and 120.)

[25] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna. The query-flow graph: Model and applications. In *CIKM '08*, 2008. (Cited on pages 116, 118, 122, 125, and 128.)

[26] F. Bonchi, R. Perego, F. Silvestri, H. Vahabi, and R. Venturini. Efficient query recommendations in the long tail via center-piece subgraphs. In *SIGIR '12*, 2012. (Cited on pages 116 and 118.)

[27] L. Bonnefoy, V. Bouvier, and P. Bellot. A weakly-supervised detection of entity central documents in a stream. In *SIGIR '13*. ACM, 2013. (Cited on page 25.)

[28] A. Bordes, J. Weston, R. Collobert, and Y. Bengio. Learning structured embeddings of knowledge bases. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, 2011. (Cited on page 28.)

[29] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26*, 2013. (Cited on page 29.)

[30] A. Bordes, X. Glorot, J. Weston, and Y. Bengio. A semantic matching energy function for learning with multi-relational data. In *Machine Learning: Special Issue on Learning Semantics*, 2014. (Cited on page 28.)

[31] I. Bordino, G. De Francisci Morales, I. Weber, and F. Bonchi. From Machu_Picchu to "rafting the urubamba river": Anticipating information needs via the entity-query graph. *WSDM '13: 6th International Conference on Web Search and Data Mining*, pages 275–284, 2013. (Cited on pages 46 and 128.)

[32] I. Bordino, Y. Mejova, and M. Lalmas. Penguins in sweaters, or serendipitous entity search on user-generated content. *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management - CIKM '13*, pages 109–118, 2013. (Cited on page 46.)

[33] A. E. Borthwick. *A Maximum Entropy Approach to Named Entity Recognition*. PhD thesis, New York University, New York, NY, USA, 1999. (Cited on page 15.)

[34] E. Boschee, R. Weischedel, and A. Zamanian. Automatic information extraction. *ICIA '05*, pages 2–4, 2005. (Cited on page 101.)

[35] S. Brin. Extracting patterns and relations from the world wide web. In *Selected Papers from the International Workshop on The World Wide Web and Databases*, WebDB '98, 1998. (Cited on pages 27 and 54.)

[36] M. Bron, K. Balog, and M. de Rijke. Ranking related entities: components and analyses. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1079–1088, New York, NY, USA, 2010. ACM. (Cited on pages 3 and 44.)

[37] M. Bron, B. Huurnink, and M. de Rijke. Linking archives using document enrichment and term selection. In *TPDL 2011*, pages 360–371, Berlin, 2011. Springer. (Cited on page 63.)

[38] R. C. Bunescu and R. J. Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, 2005. (Cited on page 26.)

[39] R. C. Bunescu and R. J. Mooney. Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'07)*, Prague, Czech Republic, June 2007. (Cited on pages 66 and 67.)

[40] A. Burman, A. Jayapal, S. Kannan, A. Kavilikatta, Madhu abd Alhelbawy, L. Derczynski, and R. Gauzauskas. USFD at KBP 2011: Entity linking, slot filling and temporal bounding. In *Proceedings of the TAC-KBP 2011 Workshop*. NIST, 2011. (Cited on page 66.)

[41] L. Byrne and J. Dunnion. UCD IIRG at tac 2011. In *Proceedings of the TAC-KBP 2011 Workshop*. NIST, 2011. (Cited on page 66.)

[42] F. Cai, R. Reinanda, and M. de Rijke. Diversifying query auto-completion. *ACM Transactions on Information Systems*, 34(4):Article 25, October 2016. (Cited on page 10.)

[43] R. Cai, H. Wang, and J. Zhang. Learning entity representation for named entity disambiguation. *ACL 2013*, pages 267–278, 2015. (Cited on page 36.)

[44] I. Cano, I. Cs, W. Edu, and G. Cs. Distributed non-parametric representations for vital filtering: UW at TREC KBA 2014. In *TREC 2014*. NIST, 2014. (Cited on page 25.)

[45] Y. Cao, J. Li, X. Guo, S. Bai, H. Ji, and J. Tang. Name list only? target entity disambiguation in short texts. *EMNLP '15*, pages 654–664, 2015. (Cited on page 20.)

[46] D. Ceccarelli, C. Lucchese, S. Orlando, R. Perego, and S. Trani. Learning relatedness measures for entity linking. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, CIKM '13, pages 139–148, 2013. (Cited on pages 34 and 36.)

[47] E. Charton, M.-J. Meurs, L. Jean-Louis, and M. Gagnon. Mutual disambiguation for entity linking. *The 52nd Annual Meeting of ACL*, pages 476–481, 2014. (Cited on page 36.)

[48] D. L. Chaudhari, O. P. Damani, and S. Laxman. Lexical co-occurrence, statistical significance, and word association. In *EMNLP '11*, pages 1058–1068, Stroudsburg, PA, USA, 2011. ACL. (Cited on page 53.)

[49] J. C. K. Cheung and X. Li. Sequence clustering and labeling for unsupervised query intent discovery. In *WSDM '12*, 2012. (Cited on page 117.)

[50] A. Chuklin, P. Serdyukov, and M. de Rijke. Using intent information to model user behavior in diversified search. In *ECIR '13*, 2013. (Cited on page 117.)

[51] M. Collins and Y. Singer. Unsupervised models for named entity classification. In *In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 100–110, 1999. (Cited on page 15.)

[52] L. D. Corro, A. Abujabal, R. Gemulla, and G. Weikum. Finet: Context-aware fine-grained named entity typing. In *EMNLP '15*, 2015. (Cited on page 17.)

[53] A. Cucchiarelli and P. Velardi. Unsupervised named entity recognition using syntactic and semantic contextual evidence. *Computational Linguistics*, 2001. (Cited on page 15.)

[54] A. Culotta and J. Sorensen. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, 2004. (Cited on page 26.)

[55] J. Dalton, L. Dietz, and J. Allan. Entity query feature expansion using knowledge base links. *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 365–374, 2014. (Cited on pages 2, 38, and 39.)

[56] B. Dalvi, A. Mishra, and W. W. Cohen. Hierarchical semi-supervised classification with incomplete class hierarchies. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, 2016. (Cited on pages 21 and 23.)

[57] L. Dietz and J. Dalton. UMass at TREC 2013 knowledge base acceleration track. In *TREC 2013*. NIST, 2013. (Cited on pages 24 and 97.)

[58] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel. The automatic content extraction (ACE) program: tasks, data, and evaluation. *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC-2004)*, 2004. (Cited on pages 15, 26, and 101.)

[59] L. Dong, F. Wei, H. Sun, M. Zhou, and K. Xu. A hybrid neural model for type classification of entity mentions. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, 2015. (Cited on pages 16 and 17.)

[60] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, 2014. (Cited on pages 31 and 32.)

[61] X. L. Dong, E. Gabrilovich, G. Heitz, W. Horn, K. Murphy, S. Sun, and W. Zhang. From data fusion to knowledge fusion. *Proceedings of VLDB Endowment*, 2014. (Cited on page 32.)

[62] X. L. Dong, E. Gabrilovich, K. Murphy, V. Dang, W. Horn, C. Lugaresi, S. Sun, and W. Zhang. Knowledge-based trust: Estimating the trustworthiness of web sources. *Proceedings of VLDB Endowment*, 2015. (Cited on page 32.)

[63] S. Dutta and G. Weikum. C 3 EL : A joint model for cross-document co-reference resolution and entity linking. *EMNP '15*, pages 846–856, 2015. (Cited on page 36.)

[64] M. Efron, C. Willis, and G. Sherman. Learning sufficient queries for entity filtering. *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1091–1094, 2014. (Cited on page 24.)

[65] J. Ellis, J. Getman, and S. Strassel. Overview of linguistic resources for the TAC KBP 2014 evaluations: planning, execution, and results. In *TAC*. LDC, 2014. (Cited on pages 18 and 34.)

[66] D. K. Elson, N. Dames, and K. R. McKeown. Extracting social networks from literary fiction. In *ACL '10*, pages 138–147, Stroudsburg, PA, USA, 2010. ACL. (Cited on page 53.)

[67] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Web-scale information extraction in knowitall: (preliminary results). In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, 2004. (Cited on page 27.)

[68] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and Mausam. Open information extraction: The second generation. *IJCAI 2011*, pages 3–10, 2011. (Cited on page 100.)

[69] A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. *EMNLP '11 Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, 2011. (Cited on page 100.)

[70] L. Fang, A. D. Sarma, C. Yu, and P. Bohannon. Rex: Explaining relationships between entity pairs. *Proc.*

VLDB Endow., 2011. (Cited on pages 48 and 49.)

[71] G. Farkas. *Essays on Elite Networks in Sweden: Power, social integration, and informal contacts among political elites*. PhD thesis, Stockholm University, 2012. (Cited on page 51.)

[72] H. Feild and J. Allan. Task-aware query recommendation. In *SIGIR '13*, 2013. (Cited on page 118.)

[73] P. Ferragina and U. Scaiella. Tagme: On-the-fly annotation of short text fragments (by Wikipedia entities). In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, 2010. (Cited on page 36.)

[74] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL '05*, pages 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. (Cited on pages 15 and 55.)

[75] S. Fissaha Adafre and M. de Rijke. Discovering missing links in Wikipedia. In *Proceedings of the Workshop on Link Discovery: Issues, Approaches and Applications (LinkKDD-2005)*. ACM, August 2005. (Cited on page 99.)

[76] L. Flekova, O. Ferschke, and I. Gurevych. What makes a good biography? *Proceedings of the 23rd International Conference on World Wide Web - WWW '14*, pages 855–866, 2014. (Cited on page 32.)

[77] J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, F. Niu, Z. Ce, R. Christopher, and I. Soboroff. Building an entity-centric stream filtering test collection for TREC 2012. In *TREC 2012*. NIST, 2012. (Cited on pages 24 and 97.)

[78] J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, E. Voorhees, and I. Soboroff. Trec kba overview. In *TREC 2014*. NIST, 2014. (Cited on pages 4, 25, 105, and 107.)

[79] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29: 1189–1232, 2000. (Cited on page 104.)

[80] A. Fuxman. In situ insights. *SIGIR '15*, pages 655–664, 2015. (Cited on pages 4 and 47.)

[81] O.-E. Ganea, M. Horlescu, A. Lucchi, C. Eickhoff, and T. Hofmann. Probabilistic bag-of-hyperlinks model for entity linking. *WWW '16*, 2015. (Cited on page 36.)

[82] J. Gao, P. Pantel, M. Gamon, X. He, and L. Deng. Modeling interestingness with deep neural networks. *EMNLP '14 Proceedings of Conference on Empirical Methods in Natural Language Processing*, 2014. (Cited on page 46.)

[83] M. Gardner and T. Mitchell. Efficient and expressive knowledge base completion using subgraph feature extraction. *Proceedings of EMNLP*, pages 1488–1498, 2015. (Cited on page 30.)

[84] M. Gardner, P. P. Talukdar, B. Kisiel, and T. Mitchell. Improving learning and inference in a large knowledge-base using latent syntactic cues. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 833–838, 2013. (Cited on page 30.)

[85] G. Garrido, A. Peñas, B. Cabaleiro, and A. Rodrigo. Temporally anchored relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 107–116, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. (Cited on pages 3 and 66.)

[86] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006. (Cited on page 56.)

[87] D. Graus, M. Tsagkias, L. Buitinck, and M. de Rijke. Generating pseudo-ground truth for predicting new concepts in social streams. In *36th European Conference on Information Retrieval*, ECIR '14, 2014. (Cited on page 19.)

[88] D. Graus, M. Tsagkias, W. Weerkamp, E. Meij, and M. de Rijke. Dynamic collective entity representations for entity ranking. In *The 9th International Conference on Web Search and Data Mining*, WSDM 2016, 2016. (Cited on page 43.)

[89] D. Graus, D. Odijk, and M. de Rijke. The birth of collective memories: Analyzing emerging entities in text streams. *Journal of the Association for Information Science and Technology*, 2017. (Cited on page 18.)

[90] R. Grishman and B. Sundheim. Message Understanding Conference-6: A brief history. *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, 1996. (Cited on page 14.)

[91] Y. Guo, B. Qin, T. Liu, and S. Li. Microblog entity linking by leveraging extra posts. In *EMNLP '13*, number October in EMNLP '13, pages 863–868, 2013. (Cited on page 36.)

[92] Z. GuoDong, S. Jian, Z. Jie, and Z. Min. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, 2005. (Cited on page 26.)

[93] D. K. Harman and E. M. Voorhees, editors. *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press, 2005. (Cited on page 57.)

[94] F. Hasibi, K. Balog, and S. E. Bratsberg. Entity linking in queries: Tasks and evaluation. In *Proceedings*

of the 2015 International Conference on The Theory of Information Retrieval, ICTIR '15, 2015. (Cited on page 79.)

[95] F. Hasibi, K. Balog, and S. E. Bratsberg. Exploiting entity linking in queries for entity retrieval. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, ICTIR '16, 2016. (Cited on page 42.)

[96] A. Hassan Awadallah, R. W. White, P. Pantel, S. T. Dumais, and Y.-M. Wang. Supporting complex search tasks. In *CIKM '14*, 2014. (Cited on pages 116 and 118.)

[97] M. Hegde. An entity-centric approach for overcoming knowledge graph sparsity. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, number September in EMNLP '15, pages 530–535, 2015. (Cited on page 31.)

[98] S. Heindorf, M. Potthast, B. Stein, and G. Engels. Towards vandalism detection in knowledge bases: Corpus construction and analysis. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, 2015. (Cited on page 33.)

[99] S. Heindorf, M. Potthast, B. Stein, and G. Engels. Vandalism detection in Wikidata. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, 2016. (Cited on page 33.)

[100] J. Hicks, V. A. Traag, and R. Reinanda. Old questions, new techniques: A research note on the computational identification of political elites. *Comparative Sociology*, 14(3):386–401, aug 2015. (Cited on page 10.)

[101] J. Hicks, V. A. Traag, and R. Reinanda. Turning digitised newspapers into networks of political elites. *Asian Journal of Social Science*, 43(5):567–587, jan 2015. (Cited on page 10.)

[102] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, 2011. (Cited on page 36.)

[103] J. Hoffart, Y. Altun, and G. Weikum. Discovering emerging entities with ambiguous names. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, 2014. (Cited on pages 19 and 20.)

[104] J. Hoffart, D. Milchevski, G. Weikum, A. Anand, and J. Singh. The knowledge awakens: Keeping knowledge bases fresh with emerging entities. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, 2016. (Cited on page 23.)

[105] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, 2011. (Cited on pages 27 and 67.)

[106] L. Hollink, P. Mika, and R. Blanco. Web usage mining with semantic analysis. In *WWW '13*, 2013. (Cited on page 115.)

[107] D. Hovy. How well can we learn interpretable entity types from text? *ACL*, 2014. (Cited on page 23.)

[108] D. Hovy, C. Zhang, E. Hovy, and A. Peñas. Unsupervised discovery of domain-specific knowledge from text. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, 2011. (Cited on page 23.)

[109] Y. Hu, Y. Qian, H. Li, D. Jiang, J. Pei, and Q. Zheng. Mining query subtopics from search log data. In *SIGIR '12*, 2012. (Cited on page 117.)

[110] R. Jenatton, N. L. Roux, A. Bordes, and G. Obozinski. A latent factor model for highly multi-relational data. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, NIPS'12, 2012. (Cited on page 28.)

[111] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao. Knowledge graph embedding via dynamic mapping matrix. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 687–696, 2015. (Cited on page 29.)

[112] H. Ji, R. Grishman, and H. T. Dang. Overview of the TAC 2011 knowledge base population task. In *Proceedings of the TAC-KBP 2011 Workshop*. NIST, 2011. (Cited on pages 3, 65, and 71.)

[113] H. Ji, T. Cassidy, Q. Li, and S. Tamang. Tackling representation, annotation and classification challenges for temporal knowledge base population. *Knowledge and Information Systems*, pages 1–36, 2013. (Cited on page 66.)

[114] J. Jiang and C.-y. Lin. MSR KMG at TREC 2014 KBA track vital filtering task. In *TREC 2014*, 2014. (Cited on pages 25, 107, and 112.)

[115] Z. Jiang, L. Ji, J. Zhang, J. Yan, P. Guo, and N. Liu. Learning open-domain comparable entity graphs from user search queries. *Proceedings of the 22nd ACM International Conference Information and*

Knowledge Management - CIKM '13, pages 2339–2344, 2013. (Cited on page 20.)

[116] T. Joachims. Training linear SVMs in linear time. In *KDD '06*, pages 217–226, New York, NY, USA, 2006. ACM. (Cited on pages 52 and 56.)

[117] R. Jones and K. L. Klinkner. Beyond the session timeout: Automatic hierarchical segmentation of search topics in query logs. In *CIKM '08*, 2008. (Cited on page 118.)

[118] N. Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*, ACL '04, 2004. (Cited on page 26.)

[119] C. Kang, S. Vadrevu, R. Zhang, R. v. Zwol, L. G. Pueyo, N. Torzec, J. He, and Y. Chang. Ranking related entities for web search queries. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, 2011. (Cited on pages 4, 45, and 79.)

[120] H. Kautz, B. Selman, and M. Shah. Referral web: combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, 1997. (Cited on page 53.)

[121] D. I. Kim, P. K. Gopalan, D. Blei, and E. Sudderth. Efficient online inference for bayesian nonparametric relational models. In *NIPS'13*, 2013. (Cited on page 88.)

[122] D. Klein and C. D. Manning. Accurate unlexicalized parsing. In *ACL '03*, pages 423–430, Stroudsburg, PA, USA, 2003. ACL. (Cited on page 54.)

[123] B. Kotnis, P. Bansal, and P. Talukdar. Knowledge base inference using bridging entities. *Proceedings of EMNLP*, pages 2038–2043, 2015. (Cited on page 30.)

[124] S. Kripke. Naming and necessity, 1980. (Cited on page 11.)

[125] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of Wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, 2009. (Cited on page 19.)

[126] N. Lao and W. W. Cohen. Relational retrieval using a combination of path-constrained random walks. *Machine Learning*, 2010. (Cited on pages 4, 29, 31, 80, and 94.)

[127] J. Lee, A. Fuxman, B. Zhao, and Y. Lv. Leveraging knowledge bases for contextual entity exploration. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1949–1958. ACM, 2015. (Cited on pages 4 and 46.)

[128] T. Lei, R. Barzilay, and T. Jaakkola. Rationalizing neural predictions. In *Empirical Methods in Natural Language Processing*, 2016. (Cited on page 143.)

[129] L. Li, H. Deng, A. Dong, Y. Chang, and H. Zha. Identifying and labeling search tasks via query-based hawkes processes. In *SIGKDD '14*, 2014. (Cited on page 118.)

[130] Q. Li, J. Artiles, T. Cassidy, and H. Ji. Combining flat and structured approaches for temporal slot filling or: how much to compress? In *Proceedings of the 13th international conference on Computational Linguistics and Intelligent Text Processing*, CICLing'12, pages 194–205, Berlin, Heidelberg, 2012. Springer-Verlag. (Cited on pages 3, 66, 69, and 75.)

[131] X. Li, J. Tang, T. Wang, Z. Luo, and M. de Rijke. Automatically assessing Wikipedia article quality by exploiting article-editor networks. In *Proceedings of the 37th European Conference on Information Retrieval*, ECIR '15. Springer, 2015. (Cited on page 33.)

[132] Y. Li, B.-J. P. Hsu, and C. Zhai. Unsupervised identification of synonymous query intent templates for attribute intents. In *CIKM '13*, 2013. (Cited on page 117.)

[133] Y. Li, B.-J. P. Hsu, C. Zhai, and K. Wang. Mining entity attribute synonyms via compact clustering. In *CIKM '13*, 2013. (Cited on page 117.)

[134] Z. Liao, Y. Song, L.-w. He, and Y. Huang. Evaluating the effectiveness of search task trails. In *WWW '12*, 2012. (Cited on page 118.)

[135] T. Lin, Mausam, and O. Etzioni. No noun phrase left behind: Detecting and typing unlinkable entities. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, 2012. (Cited on pages 19 and 22.)

[136] T. Lin, P. Pantel, M. Gamon, A. Kannan, and A. Fuxman. Active objects: Actions for entity-centric search. In *WWW '12*, 2012. (Cited on pages 1, 79, 115, and 117.)

[137] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, 2015. (Cited on page 29.)

[138] X. Ling and D. S. Weld. Fine-grained entity recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, 2012. (Cited on pages 1, 16, and 17.)

[139] D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 1989. (Cited on page 86.)

[140] Q. Liu, L. Jiang, M. Han, Y. Liu, and Z. Qin. Hierarchical random walk inference in knowledge graphs.

In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, 2016. (Cited on page 30.)

[141] X. Liu, J. Darko, and H. Fang. A related entity based approach for knowledge base acceleration. In *TREC 2013*. NIST, 2013. (Cited on page 24.)

[142] X. Liu, Y. Li, H. Wu, M. Zhou, F. Wei, and Y. Lu. Entity Linking for Tweets. *ACL '13*, pages 1304–1311, 2013. (Cited on page 97.)

[143] C. Lucchese, S. Orlando, R. Perego, F. Silvestri, and G. Tolomei. Identifying task-based sessions in search engine query logs. In *WSDM '11*, 2011. (Cited on page 118.)

[144] C. Lucchese, S. Orlando, R. Perego, F. Silvestri, and G. Tolomei. Discovering tasks from search engine query logs. *ACM Transactions on Information Systems*, 31(3):14:1–14:43, August 2013. (Cited on page 118.)

[145] C. Lucchese, S. Orlando, R. Perego, F. Silvestri, and G. Tolomei. Modeling and predicting the task-by-task behavior of search engine users. In *OAIR '13*, 2013. (Cited on pages 116 and 118.)

[146] P. Lunenfeld, A. Burdick, J. Drucker, T. Presner, and J. Schnapp. *Digital Humanities*. MIT Press, 2012. (Cited on pages 3 and 51.)

[147] G. Luo, X. Huang, C.-Y. Lin, and Z. Nie. Joint entity recognition and disambiguation. *Proceedings of EMNLP*, pages 879–888, 2015. (Cited on page 37.)

[148] Y. Luo, Q. Wang, B. Wang, and L. Guo. Context-dependent knowledge graph embedding. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1656–1661, 2015. (Cited on page 29.)

[149] A. McCallum and W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, 2003. (Cited on page 15.)

[150] O. Medelyan, I. H. Witten, and D. Mile. Topic indexing with Wikipedia. In *WIKAI '08*, WIKAI 2008, 2008. (Cited on page 34.)

[151] E. Meij, M. Bron, L. Hollink, B. Huurnink, and M. de Rijke. Mapping queries to the linking open data cloud: A case study using DBpedia. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(4):418–433, 2011. (Cited on page 120.)

[152] E. Meij, W. Weerkamp, and M. de Rijke. Adding semantics to microblog posts. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, 2012. (Cited on pages 115 and 120.)

[153] E. Meij, K. Balog, and D. Odijk. Entity linking and retrieval. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, 2013. (Cited on page 11.)

[154] Y. Merhav, F. Mesquita, D. Barbosa, W. G. Yee, and O. Frieder. Extracting information networks from the blogosphere. *ACM Trans. Web*, 6(3):11:1–11:33, 2012. (Cited on page 53.)

[155] R. Mihalcea and A. Csomai. Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, 2007. (Cited on page 34.)

[156] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *NIPS '13*, pages 1–9, 2013. (Cited on page 120.)

[157] I. Miliaraki and R. Blanco. From Selena Gomez to Marlon Brando: Understanding explorative entity search. *International World Wide Web Conference*, pages 765–775, 2015. (Cited on pages 47 and 79.)

[158] D. Milne and I. H. Witten. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *AAAI 2008*, 2008. (Cited on page 53.)

[159] D. Milne and I. H. Witten. Learning to link with Wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, 2008. (Cited on page 34.)

[160] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, ACL '09, 2009. (Cited on pages 3, 18, 27, 54, 65, and 67.)

[161] H. Mohapatra, S. Jain, and S. Chakrabarti. Joint bootstrapping of corpus annotations and entity types. *Empirical Methods in Natural Language Processing*, pages 436–446, 2013. (Cited on pages 22 and 37.)

[162] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, 1999. (Cited on page 84.)

[163] N. Nakashole, T. Tylenda, and G. Weikum. Fine-grained semantic typing of emerging entities. *ACL*, pages 1488–1497, 2013. (Cited on pages 20 and 21.)

[164] M. Nickel, V. Tresp, and H.-P. Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning*, ICML '11, 2011. (Cited on page 27.)

[165] M. Nickel, V. Tresp, and H.-P. Kriegel. Factorizing YAGO: Scalable machine learning for linked data. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, 2012. (Cited on page 27.)

[166] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich. A review of relational machine learning for knowledge graphs. *arXiv*, 2015. (Cited on page 26.)

[167] F. Nikolaev, A. Kotov, and N. Zhiltsov. Parameterized fielded term dependence models for ad-hoc entity retrieval from knowledge graph. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, 2016. (Cited on page 42.)

[168] D. Odijk, E. Meij, I. Sijaranamual, and M. de Rijke. Dynamic query modeling for related content finding. In *SIGIR 2015: 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015. (Cited on page 47.)

[169] P. Pantel and A. Fuxman. Jigs and lures: Associating web queries with structured entities. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, 2011. (Cited on pages 1, 37, 97, and 115.)

[170] P. Pantel, T. Lin, and M. Gamon. Mining entity types from query logs via user intent modeling. In *ACL'12*, 2012. (Cited on pages 115 and 116.)

[171] M. Pasca and B. Van Durme. What you seek is what you get: Extraction of class attributes from query logs. In *IJCAI'07*, 2007. (Cited on page 117.)

[172] M. Pasca, D. Lin, J. Bigham, A. Lifchits, and A. Jain. Organizing and searching the world wide web of facts - step one: The one-million fact extraction challenge. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, AAAI'06, 2006. (Cited on page 15.)

[173] A. Passos, V. Kumar, and A. McCallum. Lexicon infused phrase embeddings for named entity resolution. In *CoNLL 2014*. ACL, 2014. (Cited on page 18.)

[174] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988. (Cited on pages 84, 86, 87, and 93.)

[175] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *JMLR*, pages 2825–2830, 2011. (Cited on pages 56 and 72.)

[176] G. Pirrò. Explaining and suggesting relatedness in knowledge graphs. In *The Semantic Web - ISWC 2015: 14th International Semantic Web Conference*, 2015. (Cited on page 80.)

[177] M. Potthast, B. Stein, and R. Gerling. Automatic vandalism detection in Wikipedia. In *Advances in Information Retrieval: 30th European Conference on IR Research*, ECIR '08. Springer, 2008. (Cited on page 33.)

[178] J. Pound, P. Mika, and H. Zaragoza. Ad-hoc object retrieval in the web of data. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, 2010. (Cited on pages 41, 79, and 115.)

[179] R. Prokofyev, G. Demartini, and P. Cudré-Mauroux. Effective named entity recognition for idiosyncratic web collections. *Proceedings of the 23rd International World Wide Web Conference*, pages 397–408, 2014. (Cited on page 18.)

[180] J. Pustejovsky, J. Castaño, R. Ingria, R. Saurí, R. Gaizauskas, A. Setzer, and G. Katz. TimeML: Robust specification of event and temporal expressions in text. In *Fifth International Workshop on Computational Semantics (IWCS-5*, 2003. (Cited on page 66.)

[181] P. Radhakrishnan, M. Gupta, and V. Varma. Modeling the evolution of product entities. *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 923–926, 2014. (Cited on page 20.)

[182] L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, 2011. (Cited on pages 19 and 36.)

[183] H. Raviv, O. Kurland, and D. Carmel. Document retrieval using entity-based language models. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, 2016. (Cited on pages 39 and 40.)

[184] R. Reinanda and M. de Rijke. Prior-informed distant supervision for temporal evidence classification. In *COLING '14*, pages 996–1006. ACL, August 2014. (Cited on page 9.)

[185] R. Reinanda, D. Odijk, and M. de Rijke. Exploring entity associations over time. In *SIGIR 2013 Workshop on Time-aware Information Access*, August 2013. (Cited on pages 9, 10, and 67.)

[186] R. Reinanda, E. Meij, and M. de Rijke. Mining, ranking and recommending entity aspects. In *SIGIR '15*,

pages 263–272. ACM, August 2015. (Cited on page 9.)

[187] R. Reinanda, E. Meij, and M. de Rijke. Document filtering for long-tail entities. In *25th ACM Conference on Information and Knowledge Management*, CIKM 2016. ACM, 2016. (Cited on pages 9 and 26.)

[188] R. Reinanda, E. Meij, and M. de Rijke. Knowledge graphs: An information retrieval perspective. In *a journal*, April 2017. (Cited on page 9.)

[189] R. Reinanda, J. Pantony, J. Dorando, and E. Meij. Impact-based entity recommendations from knowledge graphs. In *KDD 2017 (submitted)*. ACL, February 2017. (Cited on page 9.)

[190] X. Ren, A. El-Kishky, C. Wang, F. Tao, C. R. Voss, and J. Han. Clustype: Effective entity recognition and typing by relation phrase-based clustering. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, 2015. (Cited on page 15.)

[191] X. Ren, W. He, M. Qu, L. Huang, H. Ji, and J. Han. Afet: Automatic fine-grained entity typing by hierarchical partial-label embedding. In *EMNLP '16'*, EMNLP, 2016. (Cited on page 17.)

[192] X. Ren, W. He, M. Qu, C. R. Voss, H. Ji, and J. Han. Label noise reduction in entity typing by heterogeneous partial-label embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 2016. (Cited on page 17.)

[193] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 2016. (Cited on page 143.)

[194] S. Riedel, L. Yao, and A. McCallum. Modeling relations and their mentions without labeled text. In *Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part III*, ECML PKDD'10, 2010. (Cited on pages 27, 66, and 67.)

[195] S. Riedel, L. Yao, B. M. Marlin, and A. McCallum. Relation extraction with matrix factorization and universal schemas. In *HLT-NAACL '13*, 2013. (Cited on page 30.)

[196] E. Riloff and R. Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, AAAI '99/IAAI '99, 1999. (Cited on page 15.)

[197] A. Ritter, L. Zettlemoyer, Mausam, and O. Etzioni. Modeling missing data in distant supervision for information extraction. In *Transactions of the Association for Computational Linguistics*, TACL'13, pages 367–378, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. (Cited on page 67.)

[198] U. Sawant and S. Chakrabarti. Learning joint query interpretation and response ranking. In *WWW '13*, 2013. (Cited on page 120.)

[199] M. Schuhmacher and S. P. Ponzetto. Knowledge-based graph document modeling. *Proceedings of the 7th ACM international conference on Web search and data mining - WSDM '14*, pages 543–552, 2014. (Cited on page 2.)

[200] M. Schuhmacher, L. Dietz, and S. Paolo Ponzetto. Ranking entities for web queries through text and knowledge. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, 2015. (Cited on page 4.)

[201] S. Sekine. NYU: Description of the Japanese NE system used for MET-2. In *Proceedings of the Seventh Message Understanding Conference (MUC-7*, 1998. (Cited on page 15.)

[202] S. Sekine. *Named Entities: Recognition, classification and use*. John Benjamin Publishings, 2009. (Cited on page 11.)

[203] S. Sekine and C. Nobata. Definition, dictionaries and tagger for extended named entity hierarchy. In *LREC*, 2004. (Cited on page 15.)

[204] S. Seufert, K. Berberich, S. J. Bedathur, S. K. Kondreddi, P. Ernst, and G. Weikum. Espresso: Relationships between entity sets. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, 2016. (Cited on pages 48 and 49.)

[205] A. Sil and S. Cucerzan. Temporal scoping of relational facts based on Wikipedia data. In *CoNLL: Conference on Natural Language Learning*, 2014. (Cited on pages 3 and 66.)

[206] A. Sil and A. Yates. Re-ranking for joint named-entity recognition and linking. *Conference on Information and Knowledge Management*, pages 2369–2374, 2013. (Cited on page 36.)

[207] J. Singh, J. Hoffart, and A. Anand. Discovering entities with just a little help from you. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, 2016. (Cited on page 23.)

[208] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, 2012. (Cited on

page 28.)

[209] R. Socher, D. Chen, C. D. Manning, and A. Y. Ng. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, 2013. (Cited on page 28.)

[210] W. Song, Q. Yu, Z. Xu, T. Liu, S. Li, and J.-R. Wen. Multi-aspect query summarization by composite query. In *SIGIR '12*, pages 325–334. ACM, 2012. (Cited on page 117.)

[211] D. Spina, E. Meij, M. de Rijke, A. Oghina, M. T. Bui, and M. Breuss. Identifying entity aspects in microblog posts. In *SIGIR '12*, 2012. (Cited on page 118.)

[212] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 697–706. ACM, 2007. (Cited on page 1.)

[213] M. Surdeanu, S. Gupta, J. Bauer, D. McClosky, A. X. Chang, V. I. Spitkovsky, and C. D. Manning. Stanford's distantly-supervised slot-filling system. In *Proceedings of the TAC-KBP 2011 Workshop*. NIST, 2011. (Cited on pages 3 and 66.)

[214] M. Surdeanu, J. Tibshirani, R. Nallapati, and C. D. Manning. Multi-instance multi-label learning for relation extraction. *EMNLP '12*, pages 455–465, 2012. (Cited on pages 27 and 67.)

[215] I. Sutskever, J. B. Tenenbaum, and R. R. Salakhutdinov. Modelling relational data using bayesian clustered tensor factorization. In *Advances in Neural Information Processing Systems 22*, pages 1821–1828. NIPS '09, 2009. (Cited on page 27.)

[216] I. Szpektor, A. Gionis, and Y. Maarek. Improving recommendation for long-tail queries via templates. In *WWW '11*, 2011. (Cited on pages 116, 118, and 128.)

[217] S. Tamang and H. Ji. Relabeling distantly supervised training data for temporal knowledge base population. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, AKBC-WEKEX '12, pages 25–30, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. (Cited on page 67.)

[218] C. H. Tan, E. Agichtein, P. Ipeirotis, and E. Gabrilovich. Trust, but verify: Predicting contribution quality for knowledge base construction and curation. *Proceedings of the 7th ACM international conference on Web search and data mining - WSDM '14*, pages 553–562, 2014. (Cited on page 32.)

[219] J. Tang, D. Zhang, and L. Yao. Social network extraction of academic researchers. In *ICDM '07*, pages 292–301, Washington, DC, USA, 2007. IEEE Computer Society. (Cited on page 53.)

[220] J. Tang, Z. Fang, and J. Sun. Incorporating social context and domain knowledge for entity recognition. *WWW '15*, pages 517–526, 2015. (Cited on page 18.)

[221] F. Tao, B. Zhao, A. Fuxman, Y. Li, and J. Han. Leveraging pattern semantics for extracting entities in enterprises. *WWW*, pages 1078–1088, 2015. (Cited on page 18.)

[222] E. F. Tjong Kim Sang and F. De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, 2003. (Cited on page 15.)

[223] H. Tong and C. Faloutsos. Center-piece subgraphs: Problem definition and fast solutions. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, 2006. (Cited on pages 48 and 49.)

[224] A. Tonon, G. Demartini, and P. Cudré-Mauroux. Combining inverted indices and structured search for ad-hoc object retrieval. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 125–134, 2012. (Cited on pages 41 and 42.)

[225] K. Toutanova and D. Chen. Observed versus latent features for knowledge base and text inference. In *3rd Workshop on Continuous Vector Space Models and Their Compositionality*. ACL – Association for Computational Linguistics, 2015. (Cited on page 30.)

[226] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon. Representing text for joint embedding of text and knowledge bases. In *Empirical Methods in Natural Language Processing (EMNLP)*. ACL – Association for Computational Linguistics, 2015. (Cited on page 30.)

[227] V. A. Traag, R. Reinanda, and G. van Klinken. Elite co-occurrence in the media. *Asian Journal of Social Science*, 43(5):588–612, jan 2015. (Cited on page 10.)

[228] V. A. Traag, R. Reinanda, and G. van Klinken. Structure of a media co-occurrence network. In S. Battiston, F. D. Pellegrini, G. Caldarelli, and E. Merelli, editors, *Proceedings of ECCS 2014*, pages 81–91. Springer International Publishing, 2016. (Cited on page 10.)

[229] N. UzZaman, H. Llorens, J. F. Allen, L. Derczynski, M. Verhagen, and J. Pustejovsky. TempEval-3: Evaluating events, time expressions, and temporal relations. *CoRR*, abs/1206.5333, 2012. (Cited on page 66.)

[230] C. Van Gysel, M. de Rijke, and E. Kanoulas. Learning latent vector spaces for product search. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*,

CIKM '16, 2016. (Cited on page 41.)

[231] C. Van Gysel, M. de Rijke, and M. Worring. Unsupervised, efficient and semantic expertise retrieval. In *WWW 2016: 25th International World Wide Web Conference*, 2016. (Cited on page 41.)

[232] D. Vannella, D. Jurgens, D. Scarfini, D. Toscani, and R. Navigli. Validating and Extending Semantic Knowledge Bases using Video Games with a Purpose. *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 1294–1304, 2014. (Cited on page 34.)

[233] M. Verma and E. Yilmaz. Entity oriented task extraction from query logs. In *CIKM '14*, 2014. (Cited on page 118.)

[234] N. Voskarides, E. Meij, M. Tsagkias, M. de Rijke, and W. Weerkamp. Learning to explain entity relationships in knowledge graphs. In *ACL-IJCNLP 2015*, 2015. (Cited on pages 49, 79, 95, and 97.)

[235] J. Wang, D. Song, C. Lin, and L. Liao. BIT and MSRA at TREC KBA CCR Track 2013. In *TREC 2013*. NIST, 2013. (Cited on pages 25, 98, 102, 107, and 118.)

[236] J. Wang, D. Song, Q. Wang, Z. Zhang, L. Si, L. Liao, and C.-Y. Lin. An entity class-dependent discriminative mixture model for cumulative citation recommendation. In *SIGIR '15*, pages 635–644. ACM, 2015. (Cited on pages 24 and 98.)

[237] X. Wang, X. L. Dong, and A. Meliou. Data x-ray: A diagnostic tool for data errors. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, 2015. (Cited on page 32.)

[238] Z. Wang, J. Zhang, J. Feng, and Z. Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, 2014. (Cited on page 29.)

[239] J. Washtell and K. Markert. A comparison of windowless and window-based computational association measures as predictors of syntagmatic human associations. In *EMNLP '09*, pages 628–637, Stroudsburg, PA, USA, 2009. ACL. (Cited on page 53.)

[240] R. West, E. Gabrilovich, K. Murphy, S. Sun, R. Gupta, and D. Lin. Knowledge base completion via search-based question answering. *Proceedings of the 23rd international conference on World Wide Web - WWW '14*, pages 515–526, 2014. (Cited on page 31.)

[241] F. Wu, J. Madhavan, and A. Halevy. Identifying aspects for web-search queries. *Journal of Artificial Intelligence Research*, 40(1):677–700, 2011. (Cited on pages 117 and 131.)

[242] Z. Wu, Y. Song, and C. L. Giles. Exploring multiple feature spaces for novel entity discovery. In *AAAI 2016*, February 2016. (Cited on page 19.)

[243] C. Xiong and J. Callan. EsdRank : Connecting query and documents through external semi-structured data. *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 951–960, 2015. (Cited on page 39.)

[244] C. Xiong and J. Callan. Query expansion with freebase. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, ICTIR '15, 2015. (Cited on page 38.)

[245] W. Xu, R. Hoffmann, L. Zhao, and R. Grishman. Filling knowledge base gaps for distant supervision of relation extraction. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 665–670, 2013. (Cited on pages 66 and 67.)

[246] Y. Yaghoobzadeh and H. Schutze. Corpus-level Fine-grained Entity Typing Using Contextual Information. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 715–725, 2015. (Cited on page 21.)

[247] L. Yan, R. Dodier, M. C. Mozer, and R. Wolniewicz. Optimizing classifier performance via an approximation to the wilcoxon-mannwhitney statistic. In *ICML'03*, 2003. (Cited on page 89.)

[248] B. Yang, W. Yih, X. He, J. Gao, and L. Deng. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the International Conference on Learning Representations (ICLR) 2015*, 2015. (Cited on page 29.)

[249] Z. Yang, W. W. Cohen, and R. Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. *CoRR*, 2016. (Cited on page 40.)

[250] L. Yao, S. Riedel, and A. McCallum. Collective cross-document relation extraction without labelled data. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, 2010. (Cited on page 27.)

[251] X. Yao and B. Van Durme. Information extraction over structured data: Question answering with Freebase. *ACL '14*, pages 956–966, 2014. (Cited on page 1.)

[252] X. Yin and S. Shah. Building taxonomy of web search intents for name entity queries. In *WWW '10*, 2010. (Cited on pages 115 and 117.)

[253] D. Yogatama, D. Gillick, and N. Lazic. Embedding methods for fine grained entity type classification. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th*

International Joint Conference on Natural Language Processing, pages 291–296, 2015. (Cited on page 17.)

[254] M. A. Yosef, S. Bauer, J. Hoffart, M. Spaniol, and G. Weikum. Hyena: Hierarchical type classification for entity names. In *Proceedings of the 24th International Conference on Computational Linguistics*, COLING '12, 2012. (Cited on pages 17 and 18.)

[255] X. Yu, H. Ma, B.-j. P. Hsu, and J. Han. On building entity recommender systems using user click log and Freebase knowledge. *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, pages 263–272, 2014. (Cited on pages 2 and 45.)

[256] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 283–292, New York, NY, USA, 2014. ACM. (Cited on pages 2 and 45.)

[257] D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, 2002. (Cited on page 26.)

[258] D. Zeng, K. Liu, Y. Chen, and J. Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762, 2015. (Cited on page 27.)

[259] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W. Y. Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 2016. (Cited on page 47.)

[260] S. Zhao and R. Grishman. Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, 2005. (Cited on page 26.)

[261] N. Zhiltsov and E. Agichtein. Improving entity search over linked data by modeling latent semantics. *Proceedings of the 22nd ACM international conference on Conference on information and knowledge management - CIKM '13*, pages 1253–1256, 2013. (Cited on page 42.)

[262] N. Zhiltsov, A. Kotov, and F. Nikolaev. Fielded sequential dependence model for ad-hoc entity retrieval in the web of data. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, 2015. (Cited on page 42.)

[263] H. Zhong, J. Zhang, Z. Wang, H. Wan, and Z. Chen. Aligning knowledge and text embeddings by entity descriptions. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 267–272, 2015. (Cited on page 31.)

[264] M. Zhou and K. C.-C. Chang. Entity-centric document filtering: Boosting feature mapping through meta-features. In *CIKM '13*, pages 119–128. ACM, 2013. (Cited on page 25.)

Summary

In this thesis, we investigate the broad problem of computing entity associations for search. Specifically, we consider three types of entity association: *entity-entity*, *entity-document*, and *entity-aspect* associations. We touch upon various domains, starting with specific domains such as the humanities and business, and ending in Web search.

The first type of association is *entity-entity* association. We begin our investigation by considering entity networks as a means of exploring document collections, and formulate the task of entity network extraction as ranking related entities. We combine approaches based on association finding and relation extraction to address the task. In our second study we go in a different direction, and focus on establishing temporal boundaries between pairs of entities having confirmed the relations between them. Finally, we bring the type of the relations to the forefront in our third study. We consider the task of recommending entities related to a query entity given their direct and indirect connections in a knowledge graph. We formalize the task of impact-based entity recommendation and propose two approaches based on learning to rank and impact propagation.

Our second theme concerns *entity-document* associations. We study this type of association in the context of filtering documents for knowledge base acceleration. In this setting, the goal is to filter documents that are relevant to update a profile of an entity. We focus on the challenge of long-tail entities specifically, and propose an approach that leverages intrinsic, i.e., *in-document* signals more than extrinsic signals such as Wikipedia page views and trending queries.

Finally, we explore *entity-aspect* associations. Entities are often associated with attributes, types, distinguishing features, topics, or themes. We broadly group this type of information under the heading "aspect." We study entity aspects in the context of Web search, and define them as common search tasks in the context of an entity. Specifically, we study the problem of mining, ranking, and recommending aspects. Entity aspects and their associations are mined from query logs.

This thesis contributes new task formalizations, algorithms, and insights on computing entity associations for search. Our experimental results confirm the effectiveness of our approaches within the different settings that we consider. Insights gained from this thesis will help address entity-oriented information access challenges in various domains.

Samenvatting

Dit proefschrift gaat over personen, organisaties en locaties, collectief aangeduid met de term *entiteiten*. We richten ons op het berekenen van entiteit-associaties voor zoekopdrachten, waarbij we met entiteit-associaties de relaties tussen entiteiten en andere objecten bedoelen, zoals uitgedrukt in tekst. We bestuderen drie entiteit-associaties in het bijzonder: *entiteit-entiteit*, *entiteit-document* en *entiteit-aspect*. Dit wordt gedaan in de context van verschillende domeinen, waarbij we beginnen met specifieke geesteswetenschappelijke en zakelijke domeinen, en eindigen met het internet in het algemeen.

Het eerste type entiteits-associatie is *entiteit-entiteit*. We verkennen documentcollecties aanvankelijk aan de hand van entiteitnetwerken. In het eerste hoofdstuk modelleren we het vinden van entiteit-entiteit associaties als een *ranking*-probleem. Hierbij maken we gebruik van associatie- en relatie-extractie technieken. In het tweede hoofdstuk nemen we een andere aanpak en leggen we ons toe op het onderkennen van temporele grenzen aan de relaties tussen entiteiten. Het derde hoofdstuk, tenslotte, gaat over het type van entiteit-entiteit relaties. Het gaat hierbij om het aanraden van entiteiten op basis van een entiteit in een zoekvraag, gegeven hun directe en indirecte connecties in een *knowledge graph*. We geven een formalisatie van het impact-gebaseerd aanraden van entiteiten en we stellen twee manieren voor, aan de hand van *learning to rank* en *impact propagation*.

Het tweede gedeelte gaat over *entiteit-document* associaties. We bestuderen deze associaties in de context van *Knowledge Base Acceleration* — het selecteren van documenten die bij kunnen dragen aan de profielpagina van een entiteit. We richten ons met name op laagfrequente entiteiten, en stellen een nieuwe methode voor die gebruik maakt van intrinsieke, *in-document* signalen.

Tot slot behandelen we *entiteit-aspect* associaties. Entiteiten worden vaak in verband gebracht met bepaalde kenmerken, types, eigenschappen en onderwerpen, hier gezamenlijk aangeduid als "aspecten". We bestuderen entiteitaspecten in de context van zoekmachines, en definiëren ze als vaak voorkomende zoekopdrachten gerelateerd aan entiteiten. In het bijzonder spitsen we ons toe op extraheren, rangschikken en aanraden van aspecten. Entiteitaspecten en hun associaties worden uit query logbestanden geëxtraheerd.

Dit proefschrift bevat bijdragen op het gebied van taakformalisaties, algoritmes en het gebruik van entiteitassociaties voor zoekmachines. De resultaten van de experimenten bevestigien de effictiviteit van onze aanpakken in de verschillende behandelde scenarios. De opgedane inzichten zullen helpen om entiteit-gerichte ontsluiting van informatie in meerdere domeinen te verbeteren.

SIKS Dissertation Series

1998

1 Johan van den Akker (CWI) *DEGAS: An Active, Temporal Database of Autonomous Objects*

2 Floris Wiesman (UM) *Information Retrieval by Graphically Browsing Meta-Information*

3 Ans Steuten (TUD) *A Contribution to the Linguistic Analysis of 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 Analyis of Decision-Theoretic Networks*

9 Florian Waas (CWI) *Principles of Probabilistic Query Optimization*

10 Niels Nes (CWI) *Image Database Management System Design Considerations, Algorithms and Architecture*

11 Jonas Karlsson (CWI) *Scalable Distributed Data Structures for Database Management*

2001

1 Silja Renooij (UU) *Qualitative Approaches to Quantifying Probabilistic Networks*

2 Koen Hindriks (UU) *Agent Programming Languages: Programming with Mental Models*

3 Maarten van Someren (UvA) *Learning as problem solving*

4 Evgueni Smirnov (UM) *Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets*

5 Jacco van Ossenbruggen (VUA) *Processing Structured Hypermedia: A Matter of Style*

6 Martijn van Welie (VUA) *Task-based User Interface Design*

7 Bastiaan Schonhage (VUA) *Diva: 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 Weerdt (TUD) *Plan Merging in Multi-Agent Systems*

16 Menzo Windhouwer (CWI) *Feature Grammar Systems: Incremental Maintenance of Indexes to Digital Media Warehouses*

17 David Jansen (UT) *Extensions of Statecharts with Probability, Time, and Stochastic Timing*

18 Levente Kocsis (UM) *Learning Search Decisions*

2004

1 Virginia Dignum (UU) *A Model for Organizational Interaction: Based on Agents, Founded in Logic*

2 Lai Xu (UvT) *Monitoring Multi-party Contracts for E-business*

3 Perry Groot (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 politiële 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*

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*
8 Eelco Herder (UT) *Forward, Back and Home Again: Analyzing User Behavior on the Web*
9 Mohamed Wahdan (UM) *Automatic Formulation of the Auditor's Opinion*
10 Ronny Siebes (VUA) *Semantic Routing in Peer-to-Peer Systems*
11 Joeri van Ruth (UT) *Flattening Queries over Nested Data Types*
12 Bert Bongers (VUA) *Interactivation: Towards an e-cology of people, our technological environment, and the arts*
13 Henk-Jan Lebbink (UU) *Dialogue and Decision Games for Information Exchanging Agents*
14 Johan Hoorn (VUA) *Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change*
15 Rainer Malik (UU) *CONAN: Text Mining in the Biomedical Domain*
16 Carsten Riggelsen (UU) *Approximation Methods for Efficient Learning of Bayesian Networks*
17 Stacey Nagata (UU) *User Assistance for Multitasking with Interruptions on a Mobile Device*
18 Valentin Zhizhkun (UvA) *Graph transformation for Natural Language Processing*
19 Birna van Riemsdijk (UU) *Cognitive Agent Programming: A Semantic Approach*
20 Marina Velikova (UvT) *Monotone models for prediction in data mining*
21 Bas van Gils (RUN) *Aptness on the Web*
22 Paul de Vrieze (RUN) *Fundaments of Adaptive Personalisation*
23 Ion Juvina (UU) *Development of Cognitive Model for Navigating on the Web*
24 Laura Hollink (VUA) *Semantic Annotation for Retrieval of Visual Resources*
25 Madalina Drugan (UU) *Conditional log-likelihood MDL and Evolutionary MCMC*
26 Vojkan Mihajlovic (UT) *Score Region Algebra: A Flexible Framework for Structured Information Retrieval*
27 Stefano Bocconi (CWI) *Vox Populi: generating video documentaries from semantically annotated media repositories*
28 Borkur Sigurbjornsson (UvA) *Focused Information Access using XML Element Retrieval*

2007

1 Kees Leune (UvT) *Access Control and Service-Oriented Architectures*
2 Wouter Teepe (RUG) *Reconciling Information Exchange and Confidentiality: A Formal Approach*
3 Peter Mika (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*

2008

1 Katalin Boer-Sorbán (EUR) *Agent-Based Simulation of Financial Markets: A modular, continuous-time approach*

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 Bunningen (UT) *Context-Aware Querying: Better Answers with Less Effort*

15 Martijn van Otterlo (UT) *The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for the Markov Decision Process Framework in First-Order Domains*

16 Henriette van Vugt (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*

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 Nabukenya (RUN) *Improving the Quality of Organisational Policy Making using Collaboration Engineering*

5 Sietse Overbeek (RUN) *Bridging Supply and Demand for Knowledge Intensive Tasks: Based on Knowledge, Cognition, and Quality*

6 Muhammad Subianto (UU) *Understanding Classification*

7 Ronald Poppe (UT) *Discriminative Vision-Based Recovery and Recognition of Human Motion*

8 Volker Nannen (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: Controling Influences on Decision Making*

21 Stijn Vanderlooy (UM) *Ranking and Reliable Classification*

22 Pavel Serdyukov (UT) *Search For Expertise: Going beyond direct evidence*

23 Peter Hofgesang (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 Khiet Truong (UT) *How Does Real Affect Affect Affect Recognition In Speech?*

34 Inge van de Weerd (UU) *Advancing in Software Product Management: An Incremental Method Engineering Approach*

35 Wouter Koelewijn (UL) *Privacy en Politiegegevens: Over geautomatiseerde normatieve informatie-uitwisseling*

36 Marco Kalz (OUN) *Placement Support for Learners in Learning Networks*

37 Hendrik Drachsler (OUN) *Navigation Support for Learners in Informal Learning Networks*

38 Riina Vuorikari (OU) *Tags and self-organisation: a metadata ecology for learning resources in a multilingual context*

39 Christian Stahl (TUE, Humboldt-Universitaet zu Berlin) *Service Substitution: A Behavioral Approach Based on Petri Nets*

40 Stephan Raaijmakers (UvT) *Multinomial Language Learning: Investigations into the Geometry of Language*

41 Igor Berezhnyy (UvT) *Digital Analysis of Paintings*

42 Toine Bogers (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 Sicco 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. Multiagent Scheduling for Airport Ground Handling*

14 Milan Lovric (EUR) *Behavioral Finance and Agent-Based Artificial Markets*

15 Marijn Koolen (UvA) *The Meaning of Structure: the Value of Link Evidence for Information Retrieval*

16 Maarten Schadd (UM) *Selective Search in Games of Different Complexity*

17 Jiyin He (UvA) *Exploring Topic Structure: Coherence, Diversity and Relatedness*

18 Mark Ponsen (UM) *Strategic Decision-Making in complex games*

19 Ellen Rusman (OU) *The Mind ' s Eye on Personal Profiles*

20 Qing Gu (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) *Ontologogy driven Enterprise Information Systems Engineering*

50 Jeroen de Jong (TUD) *Heuristics in Dynamic Sceduling: a practical framework with a case study in elevator dispatching*

2013

1 Viorel Milea (EUR) *News Analytics for Financial Decision Support*

2 Erietta Liarou (CWI) *MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing*

3 Szymon Klarman (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*

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 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*

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 Tuliyano (RUN) *Combining System Dynamics with a Domain Modeling Method*

3 Sergio Raul Duarte Torres (UT) *Information Retrieval for Children: Search Behavior and Solutions*

4 Hanna Jochmann-Mannak (UT) *Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation*

5 Jurriaan van Reijsen (UU) *Knowledge Perspectives on Advancing Dynamic Capability*

6 Damian Tamburri (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 Ghijsen (UvA) *Methods and Models for the Design and Study of Dynamic Agent Organizations*

19 Vinicius Ramos (TUe) *Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support*

20 Mena Habib (UT) *Named Entity Extraction and Disambiguation for Informal Text: The Missing Link*

21 Kassidy Clark (TUD) *Negotiation and Monitoring in Open Environments*

22 Marieke Peeters (UU) *Personalized Educational Games: Developing agent-supported scenario-based training*

23 Eleftherios Sidirourgos (UvA/CWI) *Space Efficient Indexes for the Big Data Era*

24 Davide Ceolin (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 (CWI/TUD) *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 Zhemin 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 (OUN) *Real-time Multimodal Emotion Recognition in E-Learning*

31 Yakup Koç (TUD) *On Robustness of Power Grids*

32 Jerome Gard (UL) *Corporate Venture Management in SMEs*

33 Frederik Schadd (UM) *Ontology Mapping with Auxiliary Resources*

34 Victor de Graaff (UT) *Geosocial Recommender Systems*

35 Junchao 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 Crowdsourced Information on Cultural Artefacts*

10 George Karafotias (VUA) *Parameter Control for Evolutionary Algorithms*

11 Anne Schuth (UvA) *Search Engines that Learn from Their Users*

12 Max Knobbout (UU) *Logics for Modelling and Verifying Normative Multi-Agent Systems*

13 Nana Baah Gyan (VU) *The Web, Speech Technologies and Rural Development in West Africa: An ICT4D Approach*

14 Ravi Khadka (UU) *Revisiting Legacy Software System Modernization*

15 Steffen Michels (RUN) *Hybrid Probabilistic Logics: Theoretical Aspects, Algorithms and Experiments*

16 Guangliang Li (UvA) *Socially Intelligent Autonomous Agents that Learn from Human Reward*

17 Berend Weel (VUA) *Towards Embodied Evolution of Robot Organisms*

18 Albert Meroño Peñuela (VUA) *Refining Statistical Data on the Web*

19 Julia Efremova (TUe) *Mining Social Structures from Genealogical Data*

20 Daan Odijk (UvA) *Context & Semantics in News & Web Search*

21 Alejandro Moreno Célleri (UT) *From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground*

22 Grace Lewis (VU) *Software Architecture Strategies for Cyber-Foraging Systems*

23 Fei Cai (UvA) *Query Auto Completion in Information Retrieval*

24 Brend Wanders (UT) *Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach*

25 Julia Kiseleva (TUe) *Using Contextual Information to Understand Searching and Browsing Behavior*

26 Dilhan Thilakarathne (VU) *In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains*

27 Wen Li (TUD) *Understanding Geo-spatial Information on Social Media*

28 Mingxin Zhang (TUD) *Large-scale Agent-based Social Simulation - A study on epidemic prediction and control*

29 Nicolas Höning (CWI/TUD) *Peak reduction in decentralised electricity systems - Markets and prices for flexible planning*

30 Ruud Mattheij (UvT) *The Eyes Have It*

31 Mohammadreza Khelghati (UT) *Deep web content monitoring*

32 Eelco Vriezekolk (UT) *Assessing Telecommunication Service Availability Risks for Crisis Organisations*

33 Peter Bloem (UvA) *Single Sample Statistics, exercises in learning from just one example*

34 Dennis Schunselaar (TUe) *Configurable Process Trees: Elicitation, Analysis, and Enactment*

35 Zhaochun Ren (UvA) *Monitoring Social Media: Summarization, Classification and Recommendation*

36 Daphne Karreman (UT) *Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies*

37 Giovanni Sileno (UvA) *Aligning Law and Action: a conceptual and computational inquiry*

38 Andrea Minuto (UT) *MATERIALS THAT MATTER: Smart Materials meet Art & Interaction Design*

39 Merijn Bruijnes (UT) *Believable Suspect Agents: Response and Interpersonal Style Selection for an Artificial Suspect*

40 Christian Detweiler (TUD) *Accounting for Values in Design*

41 Thomas King (TUD) *Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance*

42 Spyros Martzoukos (UvA) *Combinatorial and Compositional Aspects of Bilingual Aligned Corpora*

43 Saskia Koldijk (RUN) *Context-Aware Support for Stress Self-Management: From Theory to Practice*

44 Thibault Sellam (UvA) *Automatic Assistants for Database Exploration*

45 Bram van de Laar (UT) *Experiencing Brain-Computer Interface Control*

46 Jorge Gallego Perez (UT) *Robots to Make you Happy*

47 Christina Weber (UL) *Real-time foresight: Preparedness for dynamic innovation networks*

48 Tanja Buttler (TUD) *Collecting Lessons Learned*

49 Gleb Polevoy (TUD) *Participation and Interaction in Projects. A Game-Theoretic Analysis*

50 Yan Wang (UVT) *The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains*

2017

1 Jan-Jaap Oerlemans (UL) *Investigating Cybercrime*

2 Sjoerd Timmer (UU) *Designing and Understanding Forensic Bayesian Networks using Argumentation*

3 Daniël Harold Telgen (UU) *Grid Manufacturing: A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines*

4 Mrunal Gawade (CWI) *Multi-core Parallelism in a Column-store*

5 Mahdieh Shadi (UvA) *Collaboration Behavior*

6 Damir Vandic (EUR) *Intelligent Information Systems for Web Product Search*

7 Roel Bertens (UU) *Insight in Information: from Abstract to Anomaly*

8 Rob Konijn (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 (RUN) *Memory-Based Text Correction*

16 Aleksandr Chuklin (UvA) *Understanding and Modeling Users of Modern Search Engines*

17 Daniel Dimov (LU) *Crowdsourced Online Dispute Resolution*

18 Ridho Reinanda (UvA) *Entity Associations for Search*