# Profiling Users and Knowledge Graphs on the Web

Dissertation

zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
(Dr.-Ing.)

der Technischen Fakultät
der Christian-Albrechts-Universität zu Kiel

Chifumi Nishioka

Kiel

2017

folgendes stehen:

1. Gutachter: Prof. Dr. Ansgar Scherp

2. Gutachter: Prof. Dr. York Sure-Vetter

Datum der mündlichen Prüfung: 25. September 2017

Zum Druck genehmigt: 16. Januar 2018

To my parents, who encourage me to believe in myself.

# Erklärung

Hiermit versichere ich,

1. dass diese Arbeit – abgesehen von der Beratung durch den Betreuer Ansgar Scherp – nach Inhalt und Form meine eigene ist,

2. dass Vorversionen einiger Teile dieser Arbeit bereits veröffentlicht wurden, nämlich

   - G. Grosse-Bölting, C. Nishioka, and A. Scherp. "Generic process for extracting user profiles from social media using hierarchical knowledge bases". In: *International Conference on Semantic Computing (ICSC)*. IEEE, 2015, pp. 197–200

   - G. Große-Bölting, C. Nishioka, and A. Scherp. "A comparison of different strategies for automated semantic document annotation". In: *International Conference on Knowledge Capture (K-CAP)*. ACM. 2015, No. 8

   - C. Nishioka and A. Scherp. "Temporal patterns and periodicity of entity dynamics in the Linked Open Data cloud". In: *International Conference on Knowledge Capture (K-CAP)*. ACM. 2015, No. 22

   - C. Nishioka, G. Große-Bölting, and A. Scherp. "Influence of time on user profiling and recommending researchers in social media". In: *International Conference on Knowledge Technologies and Data-driven Business (i-KNOW)*. ACM. 2015, No. 9

   - C. Nishioka, A. Scherp, and K. Dellschaft. "Comparing tweet classifications by authors' hashtags, machine learning, and human annotators". In: *International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*. IEEE. 2015, pp. 67–74

   - C. Nishioka and A. Scherp. "Profiling vs. time vs. content: What does matter for top-k publication recommendation based on Twitter profiles?" In: *Joint Conference on Digital Libraries (JCDL)*. ACM. 2016, pp. 171–180

- C. Nishioka and A. Scherp. "Keeping linked open data caches up-to-date by predicting the life-time of RDF triples". In: *International Conference on Web Intelligence (WI)*. ACM. 2017, pp. 73–80
- C. Nishioka and A. Scherp. "Analysing the evolution of knowledge graphs for the purpose of change verification". In: *IEEE International Conference on Semantic Computing (ICSC)*. IEEE. 2018

3. dass kein Teil dieser Arbeit bereits einer anderen Stelle im Rahmen eines Prüfungsverfahrens vorgelegen hat,
4. und dass diese Arbeit unter Einhaltung der Regeln guter wissenschaftlicher Praxis der Deutschen Forschungsgemeinschaft entstanden ist.

Chifumi Nishioka

# Acknowledgement

First, I would like to sincerely thank Prof. Dr. Ansgar Scherp for accepting me as a doctoral candidate and supervising this thesis. I cannot find any words that are sufficient to express my gratitude. He gave me many wonderful opportunities to learn about Semantic Web and knowledge discovery technologies. With his excellent guidance, I have learned a great deal, including how to transfer a blurred idea into a concrete research problem, how to drive a research, how to stay motivated, and how to report results. Thus, I would not be where I am today without his amazing support and enthusiastic encouragement. I have been very fortunate to work with him for over three years.

I will also like to express my sincere gratitude to my colleagues at the Leibniz Information Centre for Economics (ZBW). They provided me with a brilliant research environment and many opportunities to discuss their research. I especially appreciate Dr. Tamara Pianos, who leads the EconBiz portal in ZBW, without whose help and support I could not have conducted an online experiment about recommender systems of scientific publications – one of the most important blocks of this thesis.

I have been supported by the German Academic Exchange Service (DAAD) for three years. They have given me some wonderful opportunities to meet talented students from all over the world, which have inspired and motivated me.

I also thank my friends for their support during the writing this thesis, even with a continent between us. Finally, I would like to thank my parents and brothers (and dogs) for always giving me their selfless support and encouraging me to continue to study and research.

# Abstract

Profiling refers to the process of collecting important and useful information or patterns about someone or something. Due to the continuous growth of the web, profiling methods play an important role in different applications such as information retrieval and recommender systems.

In this thesis, we first demonstrate how knowledge graphs enhance profiling methods. Knowledge graphs are central databases for entities such as persons and locations and relations between them. In the last decade, a lot of knowledge graphs have been developed with the objective of encouraging information reuse and information discovery. Thus, we assume that knowledge graphs can assist profiling methods. To this end, we develop a novel profiling method using knowledge graphs called Hierarchical Concept Frequency-Inverse Document Frequency (HCF-IDF), which combines the strength of traditional term weighting method and semantics in a knowledge graph. HCF-IDF represents documents as a set of entities in a knowledge graph and their weights. We apply HCF-IDF to two applications. The first application is a recommender system that suggests relevant researchers based on a user's microblog postings. The second application is a recommender system of scientific publications based on microblog postings. In both applications, we could show our novel profiling method can effectively capture user interests and a topic of a document. As key result, the method can make competitive recommendations to users based on only the title data of scientific publications. Our novel method reveals entities that are not directly mentioned but relevant using the hierarchical structure of knowledge graphs. Therefore, it can cope with the sparsity of the title data.

While the knowledge graphs can assist profiling methods, we also present how profiling methods can improve the knowledge graphs. Since knowledge graphs are often maintained and changed manually, it is important to profile the dynamics of knowledge graphs in order to keep their integrity. To this end, we show two methods that enhance the integrity of knowledge graphs. The first method is a crawling strategy that keeps local copies of knowledge graphs up-to-date. A lot of applications store knowledge graphs as local copies to speed up the data access. However, as knowledge graphs on the web change over time, the local copies need to be updated to reflect these changes. We profile the dynamics of knowledge graphs using a linear regression model. The linear regression model reveals that the dynamics of knowledge graphs correlate with their content. To this end, we develop a novel crawling strategy based on the linear regression model. The experiment shows that it performs better than the state of the art. As second method, we have developed a time-aware change verification method for knowledge graphs. While users make changes on a knowledge graph, not all changes are correct due to mistakes or vandalisms. Change verification classifies each incoming change into a correct or incorrect one, in order to reduce workload of administrators who manually check the validity of a change. We profile how topological features such as node degree influence on the dynamics of a knowledge graph. The profiling result reveals that a knowledge graph follows the preferential attachment and densification law that are observed in social networks. The experiment demonstrates that the novel method using the topological features can improve automatic change verification. Therefore, profiling the dynamics contribute to the integrity of knowledge graphs.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

This introductory chapter provides an overview of this thesis. Section 1.1 first presents a general background and motivation behind this thesis. Section 1.2 provides four application scenarios that motivate profiling documents and knowledge graphs. Subsequently, Section 1.3 summarizes the contributions made in this thesis. Thereafter, Section 1.4 lists the previous publications that have built this thesis. Finally, Section 1.5 presents the structure of the thesis.

## 1.1 Motivation

*Profiling* refers to the process of collecting important and useful information or patterns about someone or something [EV03]. In the last decades, the world wide web (web or WWW, in short) has become the largest information space. Users are suffering from an information overload problem, whereby they have difficulty understanding an issue and making decision due to the huge amount of available information. Thus, profiling users is becoming increasingly important. One of the challenges of profiling users is capturing user interests without requiring users' explicit input or making users spend time on a long initial training period. In the current web, users not only consume but also produce information [Tof81; KH10]. In particular, they actively publish and exchange their thoughts and ideas on social media platforms such as Twitter and Facebook. Therefore, social media items such as microblog postings are naturally a promising source for user profiles [CNN+10]. However, information that users produce is typically short. It makes difficult for traditional profiling methods to construct user profiles based on social media items.

In the last decade, the *Semantic Web* [BHL01] (also referred to as *Web of Data*) has evolved [SBP14]. The Semantic Web is an extension of the world wide web, in which information is structured and well-defined. Whereas the traditional web is expected to be consumed not by machines but by humans, the Semantic Web enables machines to understand and automatically process information [BHL01]. In particular, a huge amount of information has been published in the form of *Linked Data* [HB11] on the Semantic Web. An important principle of Linked Data is its use of Uniform Resource Identifiers (URIs) to refer to entities such as persons or organizations and the Resource Description Framework (RDF), a standardized data format, to express those entities. Then, relations between entities are described by links and expressed as an RDF triple (triple, in short), which is composed of subject, predicate, and object. Thus, Linked Data forms a knowledge graph where nodes and edges can be interpreted as entities and relations, respectively. In practice, many knowledge graphs have been developed and widely used, such as DBpedia [ABK+07] and Wikidata [VK14]. Since these knowledge graphs have been generated with the objective of encouraging information reuse and information discovery, we assume that they can assist in profiling methods by enriching user profiles with background knowledge. Therefore, we explore how knowledge graphs support profiling process and understanding users. Our goal is to develop a novel profiling method that leverages knowledge graphs.

Although knowledge graphs can assist in profiling methods, they are continuously maintained and updated by humans. Therefore, several works have attempted to keep the integrity of knowledge graphs using crowdsourcing [AZS+13] and statistical method [PB14]. In contrast, we assume that it is possible to contribute to the integrity of knowledge graphs by profiling their data dynamics. In this thesis, the data dynamics refer to a pattern or process of changes in data. In particular, we investigate the influence of content as well as topological features, such as node degree of entities on the data dynamics. Based on the profiling results, we develop two novel methods that improve the integrity of knowledge graphs. The experiments showcase that both content features and topological features contribute.

## 1.2 Application Scenarios

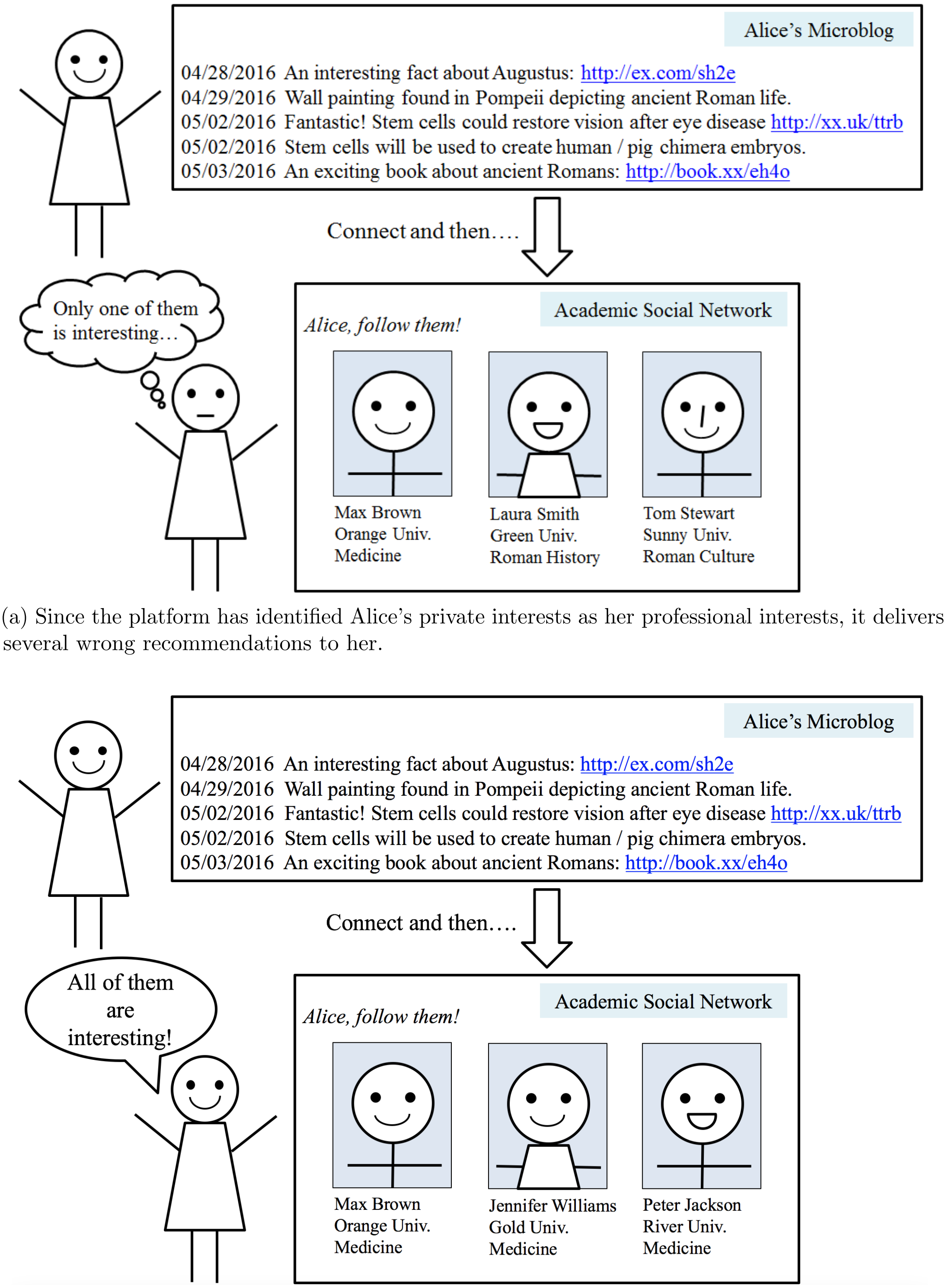
This section introduces four application scenarios that highlight challenges and motivate the research conducted for this thesis. Sections 1.2.1 and 1.2.2 motivate user profiling to understand a user's interests. Subsequently, Sections 1.2.3 and 1.2.4 demonstrate the necessity of profiling the data dynamics of knowledge graphs to maintain their integrity.

### 1.2.1 Finding Researchers

Alice is an undergraduate student in medicine. She has a microblog account and publishes microblog postings on a daily basis. Her microblog postings are not only about her private interests, e .g., history, but also about her studies. She recently started thinking about her career as a researcher. Therefore, she signed up for an academic social network platform where many researchers connect and communicate with each other. On this platform, she only knew professors at her university, so she followed them. Then, the platform suggested that she should follow several researchers. However, these researchers were only followers or followees of her professors. She later connected her microblog account to the platform, and has started to receive different recommendations of researchers to follow. For example, since she frequently posts about Roman history, which is one of her private interests, the platform has suggested researchers in the field of Roman history. In addition, she has noticed that all of her microblog postings are displayed on her page of the academic social network platform. Since almost half of her microblog postings are irrelevant to her professional interests, she would like to filter them from her page. Figure 1.1 (a) illustrates Alice's problem. To alleviate this situation, we propose a system that detects her professional interests and suggests interesting researchers based on them. In addition, this system is able to distinguish microblog postings about professional interests from those about private interests. Thus, Alice can find relevant researchers in her field on the platform as depicted in Figure 1.1 (b). It is implemented as an application and evaluated in Chapter 4.

### 1.2.2 Finding Scientific Publications

Ben is a master's student in economics. He is highly active on social media platforms: he publishes microblog postings not only about private interests (e .g., cooking, travel) but also about professional interests (e .g., economic news, interesting topics that he has learned about in class). He is currently looking for scientific publications to identify a possible research topic for his master's thesis. He could not find interesting scientific publications on a portal of the digital library. He then connected his microblog account to the portal, and it suggested several scientific publications that might interest him. However, only a few publications among them are interesting. For instance, the recommendation list contains publications relevant to "sports," which is a private interest, and "Miami," where he recently traveled. Figure 1.2 (a) depicts Ben's problem. Since the



Alice's Microblog

04/28/2016 An interesting fact about Augustus: http://ex.com/sh2e
04/29/2016 Wall painting found in Pompeii depicting ancient Roman life.
05/02/2016 Fantastic! Stem cells could restore vision after eye disease http://xx.uk/ttrb
05/02/2016 Stem cells will be used to create human / pig chimera embryos.
05/03/2016 An exciting book about ancient Romans: http://book.xx/eh4o

(a) Since the platform has identified Alice's private interests as her professional interests, it delivers several wrong recommendations to her.

Alice's Microblog

04/28/2016 An interesting fact about Augustus: http://ex.com/sh2e
04/29/2016 Wall painting found in Pompeii depicting ancient Roman life.
05/02/2016 Fantastic! Stem cells could restore vision after eye disease http://xx.uk/ttrb
05/02/2016 Stem cells will be used to create human / pig chimera embryos.
05/03/2016 An exciting book about ancient Romans: http://book.xx/eh4o

(b) A domain-specific knowledge graph helps to detect only Alice's professional interests, thus the platform can deliver right recommendations to her.

Figure 1.1: Scenario of finding researchers on an academic social network platform.

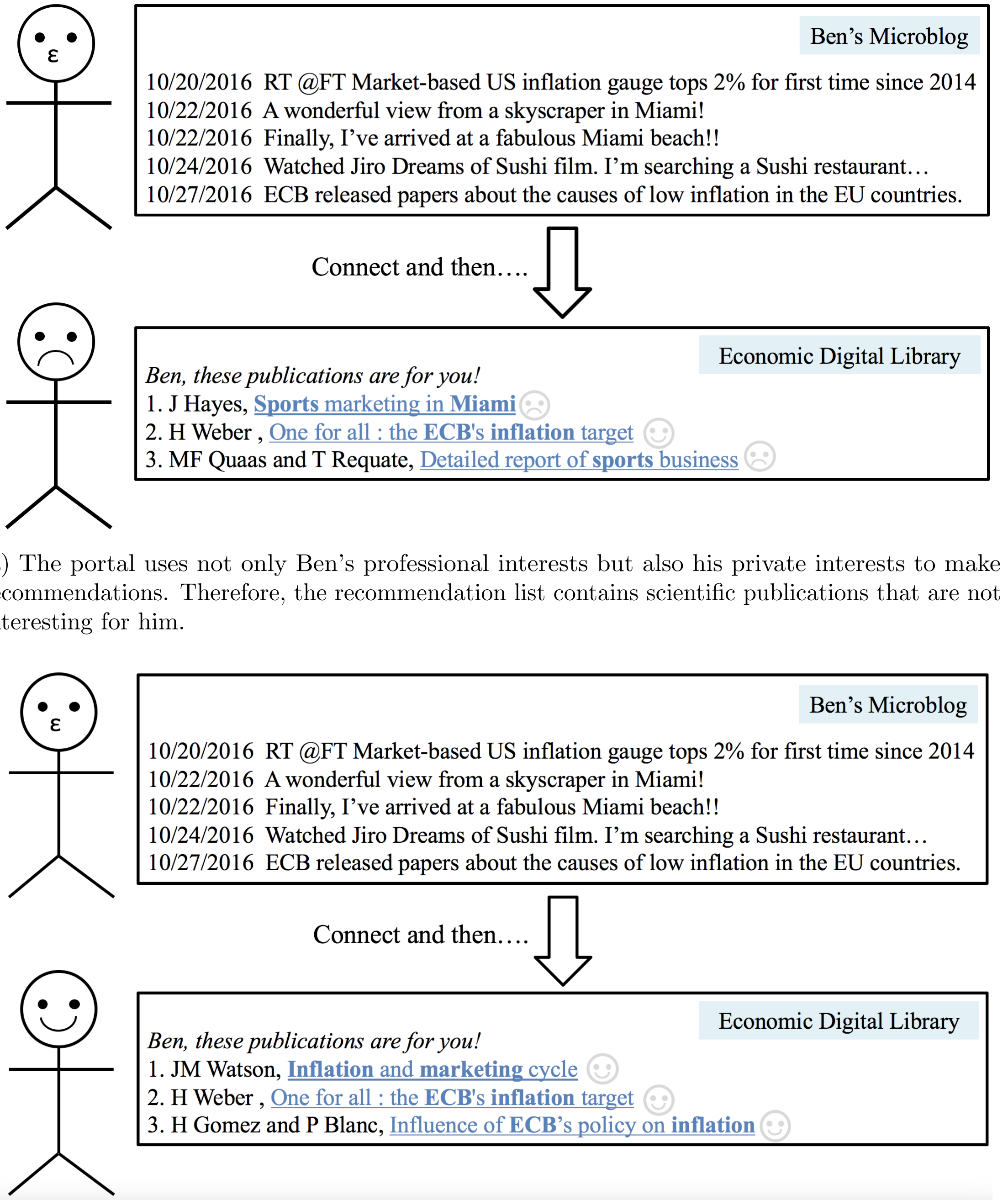
platform uses all of his tweets to extract his professional interests, only one of three recommendations is interesting for him. Thus, we have developed a recommender system that identifies his professional interests and suggests scientific publications based on them. The recommender system delivers right recommendations to him as shown in Figure 1.2 (b). Chapter 5 describes the recommender system and its evaluation.

### 1.2.3 Keeping Local Copies of Knowledge Graphs Up-to-Date

Charles develops and maintains a mobile application that uses data from knowledge graphs. These data are available as RDF documents on the web. He recently received a request from Emma, a mobile application user. Emma asked him to enable the mobile application to work even when it has no network connection because she uses the application when the connection is unstable, e .g., while running. Therefore, Charles has decided to store the RDF documents as local copies in the mobile application. However, the RDF documents on the web continuously change, so that the local copies may no longer reflect their latest state. Thus, the local copies may deliver incorrect information to the users. Charles therefore needs to implement a function that updates the local copies of the RDF documents for the mobile application. Ideally, the application would visit and download all RDF documents continuously and update their local copies. However, due to the limitations of network bandwidth and computation cost, the application cannot do this. Therefore, Charles requires an efficient crawling strategy to download the RDF documents and update their local copies to make the data as fresh as possible. This scenario is summarized in Figure 1.3. In Chapter 7, we present a novel crawling strategy that efficiently visits RDF documents and updates their local copies, thereby resolving Charles's problem.

### 1.2.4 Editing Knowledge Graphs

Dorothy is an administrator of a knowledge graph that is maintained collaboratively by editors and administrators. Editors change the information on the knowledge graph as facts in the real world change. A change is represented as an addition or deletion of a triple. Thereafter, an administrator such as Dorothy checks whether a change made by an editor is correct or incorrect. If a change is correct, it is accepted and integrated into the knowledge graph. Otherwise, it is rejected. These manual change verifications by administrators maintain the integrity of



Ben’s Microblog

10/20/2016 RT @FT Market-based US inflation gauge tops 2% for first time since 2014
10/22/2016 A wonderful view from a skyscraper in Miami!
10/22/2016 Finally, I’ve arrived at a fabulous Miami beach!!
10/24/2016 Watched Jiro Dreams of Sushi film. I’m searching a Sushi restaurant…
10/27/2016 ECB released papers about the causes of low inflation in the EU countries.

Economic Digital Library

*Ben, these publications are for you!*
1. J Hayes, **Sports** marketing in **Miami** ☹
2. H Weber , One for all : the **ECB's inflation** target ☺
3. MF Quaas and T Requate, Detailed report of **sports** business ☹

(a) The portal uses not only Ben’s professional interests but also his private interests to make recommendations. Therefore, the recommendation list contains scientific publications that are not interesting for him.

Ben’s Microblog

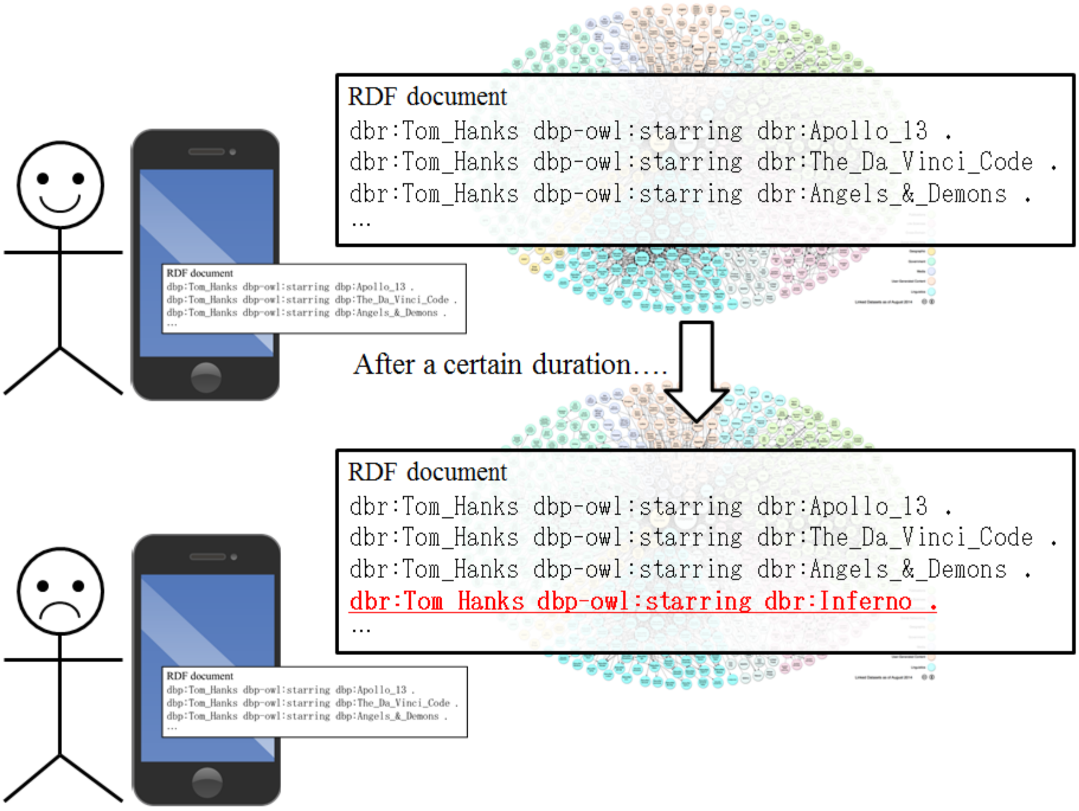
10/20/2016 RT @FT Market-based US inflation gauge tops 2% for first time since 2014
10/22/2016 A wonderful view from a skyscraper in Miami!
10/22/2016 Finally, I’ve arrived at a fabulous Miami beach!!
10/24/2016 Watched Jiro Dreams of Sushi film. I’m searching a Sushi restaurant…
10/27/2016 ECB released papers about the causes of low inflation in the EU countries.

Economic Digital Library

*Ben, these publications are for you!*
1. JM Watson, **Inflation** and **marketing** cycle ☺
2. H Weber , One for all : the **ECB's inflation** target ☺
3. H Gomez and P Blanc, Influence of **ECB’s** policy on **inflation** ☺

(b) The portal detects only Ben’s professional interests using a domain-specific knowledge graph. Thus, all recommendations are interesting for him.

Figure 1.2: Scenario of finding scientific publications in a portal of a digital library.



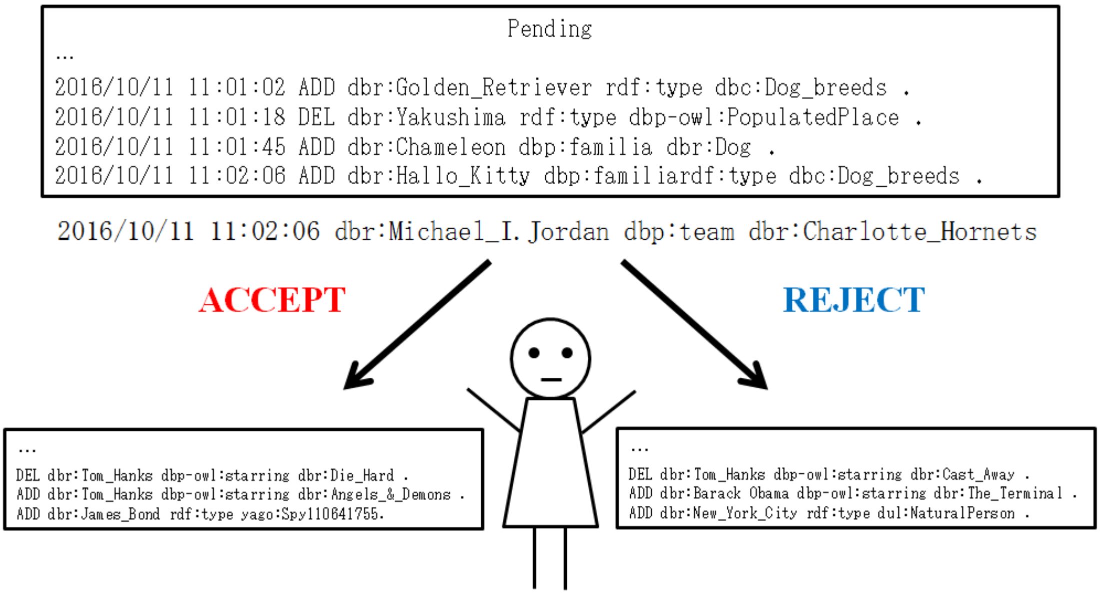
```
RDF document
dbr:Tom_Hanks dbp-owl:starring dbr:Apollo_13 .
dbr:Tom_Hanks dbp-owl:starring dbr:The_Da_Vinci_Code .
dbr:Tom_Hanks dbp-owl:starring dbr:Angels_&_Demons .
...
```

```
RDF document
dbp:Tom_Hanks dbp-owl:starring dbp:Apollo_13 .
dbp:Tom_Hanks dbp-owl:starring dbp:The_Da_Vinci_Code .
dbp:Tom_Hanks dbp-owl:starring dbp:Angels_&_Demons .
...
```

```
RDF document
dbr:Tom_Hanks dbp-owl:starring dbr:Apollo_13 .
dbr:Tom_Hanks dbp-owl:starring dbr:The_Da_Vinci_Code .
dbr:Tom_Hanks dbp-owl:starring dbr:Angels_&_Demons .
dbr:Tom_Hanks dbp-owl:starring dbr:Inferno .
...
```

```
RDF document
dbp:Tom_Hanks dbp-owl:starring dbp:Apollo_13 .
dbp:Tom_Hanks dbp-owl:starring dbp:The_Da_Vinci_Code .
dbp:Tom_Hanks dbp-owl:starring dbp:Angels_&_Demons .
...
```

Figure 1.3: Scenario of making the local copies of knowledge graphs up-to-date. The data of knowledge graphs are available as RDF documents on the web and updated there, while the local copies become stale. Thus, a crawling strategy is required that updates the local copies while respecting the limitation of network bandwidth.



```
...
2016/10/11 11:01:02 ADD dbr:Golden_Retriever rdf:type dbc:Dog_breeds .
2016/10/11 11:01:18 DEL dbr:Yakushima rdf:type dbp-owl:PopulatedPlace .
2016/10/11 11:01:45 ADD dbr:Chameleon dbp:familia dbr:Dog .
2016/10/11 11:02:06 ADD dbr:Hallo_Kitty dbp:familiardf:type dbc:Dog_breeds .
```

```
2016/10/11 11:02:06 dbr:Michael_I. Jordan dbp:team dbr:Charlotte_Hornets
```

```
...
DEL dbr:Tom_Hanks dbp-owl:starring dbr:Die_Hard .
ADD dbr:Tom_Hanks dbp-owl:starring dbr:Angels_&_Demons .
ADD dbr:James_Bond rdf:type yago:Spy110641755.
```

```
...
DEL dbr:Tom_Hanks dbp-owl:starring dbr:Cast_Away .
ADD dbr:Barack Obama dbp-owl:starring dbr:The_Terminal .
ADD dbr:New_York_City rdf:type dul:NaturalPerson .
```

Figure 1.4: Scenario of editing knowledge graphs. Administrators have to manually check whether an incoming change is correct or incorrect. As the number of incoming changes increases continuously, the administrators are overloaded by their work.

the data on the knowledge graph. While Dorothy and other administrators work industriously, however, the number of changes made by editors is increasing rapidly. Therefore, it is necessary to either hire more administrators or use a tool that facilitates the manual change verification process. The management team of the knowledge graph has indicated that it is impossible to increase the number of administrators due to cost. Therefore, Dorothy requires a tool to assist her. Figure 1.4 represents this scenario. To mitigate the overload experienced by Dorothy and other administrators, Chapter 8 proposes a novel method that automatically verifies whether an incoming change to a knowledge graph is correct or incorrect.

## 1.3 Contributions

The contributions of this thesis are summarized in the following:

- We demonstrate how knowledge graphs can support profiling documents and users. Specifically, we develop a novel profiling method, called Hierarchical Concept Frequency-Inverse Document Frequency (HCF-IDF), which is an extension of Concept Frequency-Inverse Document Frequency (CF-IDF) [GIF+11] and Term Frequency-Inverse Document Frequency (TF-

IDF) [SB88; SM86; SWY75]. The method represents a document and a user as a set of entities and their weights. Since HCF-IDF leverages the hierarchical structure of a knowledge graph, it can reveal entities that are not directly mentioned but relevant.

- We demonstrate two applications for which the profiling methods using knowledge graphs such as HCF-IDF work well. The applications are recommender systems that suggest relevant researchers and scientific publications based on a user's microblog postings. In both applications, we observe that the profiling methods using knowledge graphs make better recommendations. In addition, we find that these profiling methods work well especially for short documents, such as microblog postings and titles of scientific publications.

- We profile and understand the data dynamics on knowledge graphs. First, we investigate how the content of triples influence their life span (i .e., how long a triple is alive). We use a linear regression model to predict triples' life spans using the content of triples. The resulting model provides insights into which triples are stable and which are ephemeral. Second, we study the influence of topological features such as node degree of entities on the data dynamics of knowledge graphs. The investigation reveals that a knowledge graph follows the densification law [LKF05] and preferential attachment [New01], as observed in other graphs [LKF05]. It indicates that it is possible to predict future changes of knowledge graphs.

- We develop a novel crawling strategy for RDF documents based on our linear regression model. The existing crawling strategies [DGS15] are based on how frequently an RDF document has been modified in the past. In contrast, the novel crawling strategy predicts the data dynamics of RDF documents by considering their content. The experiment has two different settings and uses two datasets. The results demonstrate that the novel crawling strategy outperforms the state of the art.

- We present a novel method of change verification for a knowledge graph. Change verification classifies each incoming change for a knowledge graph into a correct or incorrect change. The experiment demonstrates that topological features such as node degree of entities contribute to the improvement of the classification performance.

## 1.4 Publication Record

In the last years, I have published the building blocks of this thesis at several conferences. The publications that have contributed to this thesis are listed below.

- G. Grosse-Bölting, C. Nishioka, and A. Scherp. "Generic process for extracting user profiles from social media using hierarchical knowledge bases". In: *International Conference on Semantic Computing (ICSC)*. IEEE, 2015, pp. 197–200

- G. Große-Bölting, C. Nishioka, and A. Scherp. "A comparison of different strategies for automated semantic document annotation". In: *International Conference on Knowledge Capture (K-CAP)*. ACM. 2015, No. 8

- C. Nishioka and A. Scherp. "Temporal patterns and periodicity of entity dynamics in the Linked Open Data cloud". In: *International Conference on Knowledge Capture (K-CAP)*. ACM. 2015, No. 22

- C. Nishioka, G. Große-Bölting, and A. Scherp. "Influence of time on user profiling and recommending researchers in social media". In: *International Conference on Knowledge Technologies and Data-driven Business (i-KNOW)*. ACM. 2015, No. 9

- C. Nishioka, A. Scherp, and K. Dellschaft. "Comparing tweet classifications by authors' hashtags, machine learning, and human annotators". In: *International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*. IEEE. 2015, pp. 67–74

- C. Nishioka and A. Scherp. "Profiling vs. time vs. content: What does matter for top-k publication recommendation based on Twitter profiles?" In: *Joint Conference on Digital Libraries (JCDL)*. ACM. 2016, pp. 171–180

- C. Nishioka and A. Scherp. "Keeping linked open data caches up-to-date by predicting the life-time of RDF triples". In: *International Conference on Web Intelligence (WI)*. ACM. 2017, pp. 73–80

- C. Nishioka and A. Scherp. "Analysing the evolution of knowledge graphs for the purpose of change verification". In: *IEEE International Conference on Semantic Computing (ICSC)*. IEEE. 2018

## 1.5 Outline

In the subsequent chapter, we review works related to this thesis. Chapter 3 delivers a formalization of profiling documents and users, and introduces different profiling methods using knowledge graphs. Subsequently, Chapters 4 and 5 present applications that leverage the profiling methods introduced in Chapter 3. Specifically, Chapter 4 evaluates the profiling methods in the task of recommending researchers based on a user's microblog postings, while Chapter 5 presents an online experiment on recommending scientific publications with 123 subjects. Since knowledge graphs are maintained manually, it is important to understand their data dynamics to keep their integrity. In this vein, Chapter 6 introduces different profiling methods to capture these data dynamics, and also reports profiling results. Chapters 7 and 8 demonstrate applications that make use of these data dynamics. Specifically, Chapter 7 describes a novel crawling strategy that updates the local copies of RDF documents efficiently, and Chapter 8 demonstrates a novel method of change verification for a knowledge graph. Finally, Chapter 9 reflects on the work presented in this thesis and proposes directions for future studies.

# Chapter 2

# Related Work

This chapter reviews the literature related to this thesis. These works build and support the foundation of this thesis. Section 2.1 introduces different methods for profiling documents and users. It describes traditional term-based profiling methods as well as methods supported by a knowledge graph. Subsequently, Section 2.2 discusses various profiling methods to capture data dynamics and reviews existing works on the data dynamics of knowledge graphs.

## 2.1 Profiling Users

This section reviews different methods of profiling documents and users. Figure 2.1 describes how we distinguish between documents and users. Documents such as news articles are profiled independently, while a user's set of social media items is used to produce one user profile. For user profiling, each social media item in a user's social media stream is profiled individually. Then, the profiles of the social media items are integrated into one profile. Please note that this thesis focuses on only textual content. Hence, we do not cover profiling methods for media data (e .g., images), although a document and social media item may contain them. In Section 2.1.1, we first review methods that represent profiles as a set of terms. Subsequently, we introduce methods that exploit a knowledge graph in Section 2.1.2.

### 2.1.1 Term-Based Profiling Methods

In this section, we first detail traditional term-based profiling methods. Subsequently, we describe topic modeling, which is also used as a profiling method.

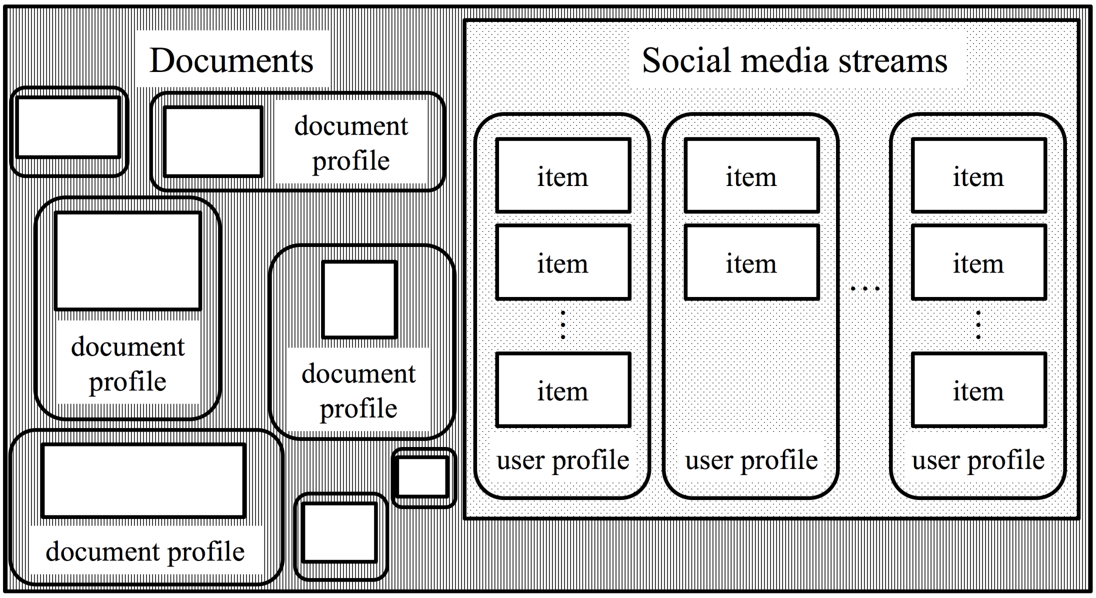


Figure 2.1: Documents and users' social media streams with their items.

**Term-based profiling methods** Term-based profiling methods analyze and profile a document by using the vector space model (VSM) [SWY75]. The VSM represents a document by a vector whose dimensions equal the number of unique terms in the document corpus. Each vector dimension corresponds to a separate term and defines the term weight, i .e., the degree of association between the document and the term. In this section, we focus on how to compute term weights.

The best-known profiling method is Term Frequency-Inverse Document Frequency (TF-IDF) [SB88; SM86; SWY75]. TF-IDF was originally introduced for information retrieval, where a document is represented as a vector of term weights. TF counts the frequency with which a term appears in a document. It is based on the assumption that multiple appearances of a term in a document are more relevant than single appearances. IDF then takes the assumption that rare terms are more important than frequent terms. Although TF-IDF [SWY75] was first introduced more than 40 years ago, it is still a robust baseline. Following TF-IDF, Okapi BM25 (BM25) [RWJ+94; RW94] was built as a probabilistic model that is sensitive to term frequency and document length [JWR00]. In addition, Rousseau and Vazirgiannis [RV13] introduced the graph-of-word model, a novel document representation, and Term Weight-Inverse Document Frequency (TW-IDF), a term weighting method. The graph-of-word model captures the relationships between terms using an unweighted directed graph of terms. Based on this graph, TW-IDF then computes a term weight. Their experiment on standard TREC datasets showed the superiority of TW-IDF compared to BM25.

In another study, Shirakawa et al. [SHN15] proposed N-gram IDF, which handles key terms of any length. N-gram IDF is based on their finding that the IDF of a term is equal to the distance between the term and the empty string in the space of information distance. Their experiment on keyword extraction and search query segmentation revealed that N-gram IDF is competitive with the state of the art methods designed for keyword extraction and search query segmentation, respectively.

TF-IDF and the other aforementioned profiling methods are used to profile users. In this line, Xu et al. [XBF+08] tackled the problem of personalized information retrieval. They used TF-IDF as well as BM25 to construct user profiles and document profiles. Moreover, Chen et al. [CNN+10] developed a recommender system for URLs (i .e., web pages) that uses TF-IDF to compute user profiles based on their microblog postings or those of their followees. The system then suggests URLs that might interest the user. Their experiment demonstrated that recommendations based on the user's microblog postings were better than recommendations based on microblog postings of his followees. In a similar vein, Phelan et al. [PMB+11] developed a recommender system for news articles in which they come from either a user's RSS space or the entire RSS space. A user profile is constructed based on either the user's microblog postings or microblog postings produced by her followees. Using TF-IDF, the recommender system make profiles of both news articles and users. The results of the experiment revealed that a user's RSS space was a better source of candidate items. In terms of the source of user profiles, microblog postings by a user's followees were slightly better than the user's own postings, although the difference was not significant. Similar to Chen et al. [CNN+10], Goossen et al. [GIF+11] developed a recommender system for news articles and used TF-IDF as their baseline. The recommender system computes user profiles based on news articles that a user has read. Moreover, Ribeiro et al. [RSG+15] aimed to profile a user's professional interests based on titles, abstracts, or/and keywords of his own scientific publications. As profiling methods, they compared term frequency, TF-IDF, and coverage [VKG11]. Their experiment with 1,288 subjects showed that term frequency and TF-IDF were significantly better than coverage.

**Topic modeling** Although the term-based VSMs discussed above perform well for different applications, including information retrieval and recommender systems, they cannot capture the hidden structure within terms in a document corpus, and have the problems of synonymy (e .g., automobile and car) and polysemy (e .g., bank referring to a financial institution and bank meaning the land alongside a body of water). To address these problems, Deerwester et al. [DDF+90] developed Latent Semantic Indexing (LSI). LSI uses singular value decomposition (SVD) to project a term-based vector representation of a document into a lower dimensional space. Compared to the term-based VSMs, LSI can achieve significant compression of a large document corpus. In addition, Deerwester et al. [DDF+90] observed that LSI could capture linguistic notions such as synonymy and polysemy. In practice, Berry et al. [BDO95] reported that the LSI improved information retrieval by addressing the problems of polysemy and synonymy. Later, Hofmann [Hof99] proposed probabilistic LSI (pLSI) as an alternative to LSI that covers its unsatisfactory statistical foundation. pLSI is based on the likelihood principle and defines a proper generative model of a document corpus. Specifically, it models each term in a document as a sample from a mixture model that can be seen as a "topic." Thus, each term is generated from a single topic, and different terms in a document are generated from different topics. However, the pLSI provides no generative model at the level of documents. Moreover, the number of parameters in the model grows linearly with the size of the corpus, which leads to overfitting. To improve this, Blei et al. [BNJ03] developed Latent Dirichlet Allocation (LDA), which provides a complete generative model for documents. The generative model specifies a simple probabilistic procedure whereby documents can be produced given a set of topics. LDA generates documents by picking a distribution over topics from a Dirichlet distribution. The terms in the document are then generated by picking a topic from this distribution, and in turn picking a term from that topic. According to Chang et al. [CBG+09], topics generated by LDA are better interpreted by humans than those by the pLSI. Although LDA was originally developed for mining documents, it has also been used to detect instructive structures in images [SRZ+08; BWP11] and genetic information [PSD00].

The above topic modeling methods have been used to profile documents and users. For example, Wang and Blei [WB11] created user profiles based on scientific publications that they read using LDA. They represented user profiles as probability distributions over topics. The user profiles were used with a recommender system for scientific publications that suggests scientific publications whose topic distribution is similar to a user profile. However, topic modeling methods such as LDA do not work well for short documents, such as microblog postings, since they rely on the co-occurrences of terms. In fact, an experiment by Hong and Davison [HD10] revealed that a topic model where microblog postings by the same user were aggregated as one document resulted in significantly better performance in two classification tasks than a topic model where each microblog posting was considered to be one document. Moreover, Pennacchiotti and Gurumurthy [PG11] used LDA to develop a recommender system that suggests friends who have similar interests based on a user's microblog postings. Similar to Hong and Davison [HD10], they aggregated microblog postings published by the same user into one document. They found that the recommendation performance of LDA outperformed TF-IDF as well as network-based methods (i .e., friends-of-friends).

Several works [RHN+09; ZJW+11; YKS+14; TMH16] have proposed variants of LDA for short documents. Ramage et al. [RHN+09] proposed Labeled LDA, motivated by the fact that a significant number of documents on the web are tagged by publishers or readers on social media platforms and social bookmarking sites, such as Delicious and Twitter hashtags. Many documents have multiple tags, but they do not have equal importance. Thus, it is necessary to associate each term in a document with the most appropriate tags and vice versa. To this end, the Labeled LDA constrains LDA by defining a one-to-one correspondence between latent topics generated by LDA and the tags. It is able to directly learn term-tag correspondences. Ramage et al. [RHN+09] showed that the Labeled LDA works well for personalized feed re-ranking and recommending new friends. Zhao et al. [ZJW+11] introduced a novel Twitter-LDA for microblog postings. It is designed based on the assumption that a single tweet contains only one topic. Furthermore, Yang et al. [YKS+14] presented a spectrum of topic modeling methods based on LDA to classify tweets in real time into a topic in a hierarchy. These methods include non-topical tweet detection, automatic labeled data acquisition, evaluation with human computation, diagnostic and corrective learning, and, most importantly, high-precision topic inference. However, although these variants of LDA work well for different tasks, they require external knowledge, the use of which may lead to bias in trained topic models.

### 2.1.2 Profiling Methods Using Knowledge Graphs

This section introduces different profiling methods that use a knowledge graph. We first introduce widely used knowledge graphs. Then, we review profiling methods that consider knowledge graphs as flat lists of entities. Beyond these profiling methods, we describe different profiling methods that exploit a hierarchical structure of a knowledge graph, and finally introduce profiling methods that can use any graph structure.

**Knowledge graph** In the last decade, many different knowledge graphs have been developed on the web and used by many applications. The best-known knowledge graph is DBpedia[1] [ABK+07]. The DBpedia project was initiated by the Free University of Berlin and the University of Leipzig, in collaboration with OpenLink Software, in 2007. DBpedia collects structured information from infoboxes, categorization, and geo-coordinates in Wikipedia articles. In addition, it also stores a large amount of external links to other datasets such as UMBEL, GeoNames, CIA World Factbook, and DBLP. Therefore, DBpedia is considered as a hub of knowledge graphs. As another example, Freebase[2] [BEP+08] was a cross-domain knowledge graph maintained by Google. It was created from inputs by editors as well as existing RDF and microformat datasets. To facilitate editors, Freebase provided an interface where editors made changes. However, Google decided to discontinue Freebase in 2015. The data from Freebase have been transferred to Wikidata[3] using the Primary Sources Tool[4] [TVS+16]. YAGO[5] is another well-known knowledge graph. It has been developed at the Max Planck Institute since 2007. It automatically extracts information about entities from Wikipedia articles, WordNet [Fel98] (e .g., synsets, hyponymy), and GeoNames. Furthermore, Wikidata [VK14] is a project of Wikimedia Deutschland that was launched in 2012. Wikidata stores not only information about entities, but also the corresponding sources. Thus, users can check the validity of information. Labels, aliases, and descriptions of entities are provided in almost 400 languages. Wikidata is collaboratively created by editors. In addition, the schema is maintained and extended based on agreements among editors. In recent years, this knowledge graph has grown rapidly due to the migration of Freebase [TVS+16]. It provides RDF exports [EGK+14].

Besides these cross-domain knowledge graphs, many different domain-specific knowledge graphs have also been developed. These are typically maintained by domain experts. Therefore, they are of high quality. For example, MeSH (Medical Subject Headings)[6] is a domain-specific knowledge graph in the field of medicine. It is maintained by National Library of Medicine (NLM) and used by PubMed article

[1] `http://wiki.dbpedia.org/`, last accessed on 08/30/2017

[2] `https://developers.google.com/freebase/`, last accessed on 08/30/2017

[3] `https://www.wikidata.org`, last accessed on 08/30/2017

[4] `https://www.wikidata.org/wiki/Wikidata:Primary_sources_tool`, last accessed on 08/30/2017

[5] `http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/`, last accessed on 08/30/2017

[6] `https://www.nlm.nih.gov/mesh/`, last accessed on 08/30/2017

database. It provides RDF exports[7] and SPARQL query interface[8]. AGROVOC[9] is a knowledge graph covering food, nutrition, agriculture, and environment. It is maintained by Food and Agriculture Organization (FAO) of United Nations. It is used by researchers and librarians for indexing and organizing documents and web pages relevant to agriculture. Other domain-specific knowledge graphs are noted in a list[10] maintained by the W3C.

All of these cross-domain and domain-specific knowledge graphs have been developed to encourage information share and reuse and facilitate information discovery. Therefore, they have been used to profile documents and users [OBP12; SP14a; NXC+16]. Below, we introduce different profiling methods that use a knowledge graph.

**Profiling methods using a list of domain-specific entities** We introduce profiling methods using a list of domain-specific entities. These profiling methods leverage entities and their labels stored in a knowledge graph, but ignore relations among them. Abel et al. [AHK11] attempted to extract a user's professional interests from social media platforms, including LinkedIn, Delicious, and Twitter. They compared the tag-based approach (for Delicious), bag-of-words approach (for LinkedIn and Twitter), and semantic entity-based approach (for Twitter) in the scenario of recommending scientific publications. They observed that Twitter seemed to cover more professional interests than other social media platforms, but also to include more noise. Regarding different profiling methods, the semantic entity-based profiles based on Twitter outperformed the others. In another study, Orlandi et al. [OBP12] built user profiles based on different social media platforms. In particular, they used Twitter and Facebook as profiling sources. They represented user profiles as sets of entities on DBpedia and their frequencies. Moreover, Goossen et al. [GIF+11] proposed Concept Frequency-Inverse Document Frequency (CF-IDF) as an extension of TF-IDF [SB88; SM86; SWY75]. CF-IDF replaces frequencies of terms with those of entities. They conducted an experiment investigating news article recommendations with 19 subjects, and found that CF-IDF outperformed TF-IDF.

**Profiling methods using hierarchical knowledge graphs** Although the profiling methods using a list of domain-specific entities have demonstrated their

[7] `ftp://ftp.nlm.nih.gov/online/mesh/rdf/`, last accessed on 08/30/2017

[8] `https://id.nlm.nih.gov/mesh/query`, last accessed on 08/30/2017

[9] `http://aims.fao.org/standards/agrovoc/linked-open-data`, last accessed on 08/30/2017

[10] `http://www.w3.org/2001/sw/wiki/SKOS/Datasets`, last accessed on 08/30/2017

effectiveness, they face challenges with short documents, such as microblog postings, because these documents usually contain only a few entities. To overcome this challenge, several profiling methods utilize the structure of a knowledge graph. Specifically, these profiling methods reveal entities that are not mentioned in a document but are relevant to it.

With the use of the hierarchical structure of a knowledge graph, Middleton et al. [MDS01] constructed user profiles based on users' browsing history and explicit feedback. Their method represented user profiles as a set of entities and their weights, and computes the latter by using a propagation function that provides 50% of weights to their broader entities. Their profiling method is the same as Basic Spreading Activation. Basic Spreading Activation is one variant of spreading activation, which is a propagation function referred to by Kapanipathi et al. [KJV+14]. Although Middleton et al. [MDS01] did not justify their choice of the value of 50%, their experiment demonstrated that their profiling method outperformed a profiling method using a list of domain-specific entities. They concluded that users preferred to have user profiles including general entities that were not directly mentioned. Kapanipathi et al. [KJV+14] further developed some profiling methods that leverage the hierarchical structure of a knowledge graph using spreading activation [CL75]. In their work, they employed their own cross-domain hierarchical knowledge graph, which is generated based on Wikipedia. User profiles are generated based on users' microblog postings. The experiment demonstrated that spreading activation enabled the creation of meaningful user profiles based on microblog postings. Moreover, Rybak et al. [RBN14] created users' professional profiles using the ACM Computing Classification System (ACM CCS) as a knowledge graph. Similar to Middleton et al. [MDS01] and Kapanipathi et al. [KJV+14], they used a propagation function that provided weights of broader entities. In their work, broader entities received 100% of the weights of their narrower entities.

**Profiling methods using knowledge graphs** There are also profiling methods that can be applied with any structure of a knowledge graph. Lu et al. [LLZ12] proposed a recommender system that suggests relevant microblog postings based on a user's own microblog postings. Their method represents user profiles as set of weighted Wikipedia entities that correspond to Wikipedia articles. It expands user profiles via random walk on the Wikipedia entity graph, which is created by utilizing the interlinks between Wikipedia articles. Their experiment demonstrated that their profiling method was effective to recommend relevant microblog postings to users. In another study, Schuhmacher and Ponzetto [SP14b] presented a profiling method using knowledge graphs that takes into account strengths of relations between entities. Each relation in a knowledge graph is weighted with different information theoretic measures. They evaluated their profiling method using DBpedia in the tasks of entity ranking and computation of similarity scores between documents. The result of the experiment revealed that their profiling method outperformed baselines and showed competitive performance against methods designed for these specific tasks. Furthermore, Ni et al. [NXC+16] proposed a profiling method that represents a document as a graph, where nodes are entities on a knowledge graph. Entities are weighted using a closeness centrality measure that reflects their relevance to the document. The authors further presented a novel measure to compute similarity scores between two documents. This measure first represents entities as continuous vectors by neural networks. These continuous vectors are then used to accumulate pairwise similarity between pairs of entities while considering their assigned weights. An experiment evaluated their profiling method on a standard benchmark for computation of similarity scores between documents. The method outperformed the state-of-the-art methods, such as that of Schuhmacher and Ponzetto [SP14b].

Although these profiling methods demonstrated good performance for different tasks including profiling users and computing similarity between documents, we do not use them, since they are computationally expensive or require training process.

### 2.1.3 Profiling Methods that Consider Data Dynamics

Since user interests may change over time, it is not trivial to consider the temporal aspect of information when profiling users. In this section, we first note the forgetting curve that describes human memory retention. Subsequently, we introduce the works that investigate the influence of older data on profiling. Finally, we review several profiling methods that take into account data dynamics.

**The forgetting curve in psychology** A large body of work in psychology and cognitive science has investigated how human memory evolves over time. At the end of 19th century, Ebbinghaus [Ebb85; Ebb13] studied the memorization of nonsense syllables, such as 'sdh" and "pdy," by repeatedly testing himself after different time periods and recording and plotting the results of these tests. The plots demonstrated that his memory retention declined as time passed; this is known as Ebbinghaus's forgetting curve. Researchers have debated the form of the forgetting curve for over a century. For instance, Wixted and Ebbesen [WE91] showed that forgetting curves produced by a variety of procedures are often well explained by the power function. However, Anderson and Tweney [AT97] argued that Wixted and Ebbesen's result [WE91] may be an artifact of arithmetically averaging subjects' forgetting curves. They argued that forgetting curves could be better explained by the exponential function. Wixted and Ebbesen [WE97] subsequently rebutted Anderson and Tweney's argument [AT97], stating that their conclusion did not change even if they used geometric averaging, as Anderson and Tweney suggested. In addition, their analysis of individual subjects' forgetting curves revealed that the power function described these curves better than the exponential function. In another study, Averell and Heathcote [AH11] collected and analyzed data from a longitudinal experiment measuring cued recall and stem completion from 1 minute to 28 days after study. The data contained more observations per interval per subject than in previous works. The authors concluded that the exponential function provided a better fit to individual subjects' forgetting curves than the power function and the Pareto function did. In this thesis, we use the exponential function to represent the data dynamics of user interests rather than the power function, since it models forgetting curves better, according to the most recent work [AH11].

**Influence of older data** In the field of computer science, several studies have investigated the influence of older data on user profiles. De Pessemier et al. [DDD+10] divided data into ten sets by chronological order and investigated which set was the best source of a collaborative filtering. They used two datasets: provider-generated content and user-generated content. The results demonstrated that the recommendation performance of the collaborative filtering improved by extending the provider-generated content with additional older data. On the other hand, the opposite effect was observed for user-generated content. This indicates that involving older user-generated data has a negative influence. Zheng and Ip [ZI13] also evaluated the influence of data generated over different periods of time on the recommendation performance of a collaborative filtering. Their results revealed that while more recent data had a larger impact, the usefulness of older data could not be ignored as long as they were in sufficient amounts. On the other hand, the addition of insufficient amounts of old data had a negative influence. Thus, the older data has both a positive and a negative influence on user profiles. Since the influence depends on the context, it is necessary to carefully consider and examine the use of older data.

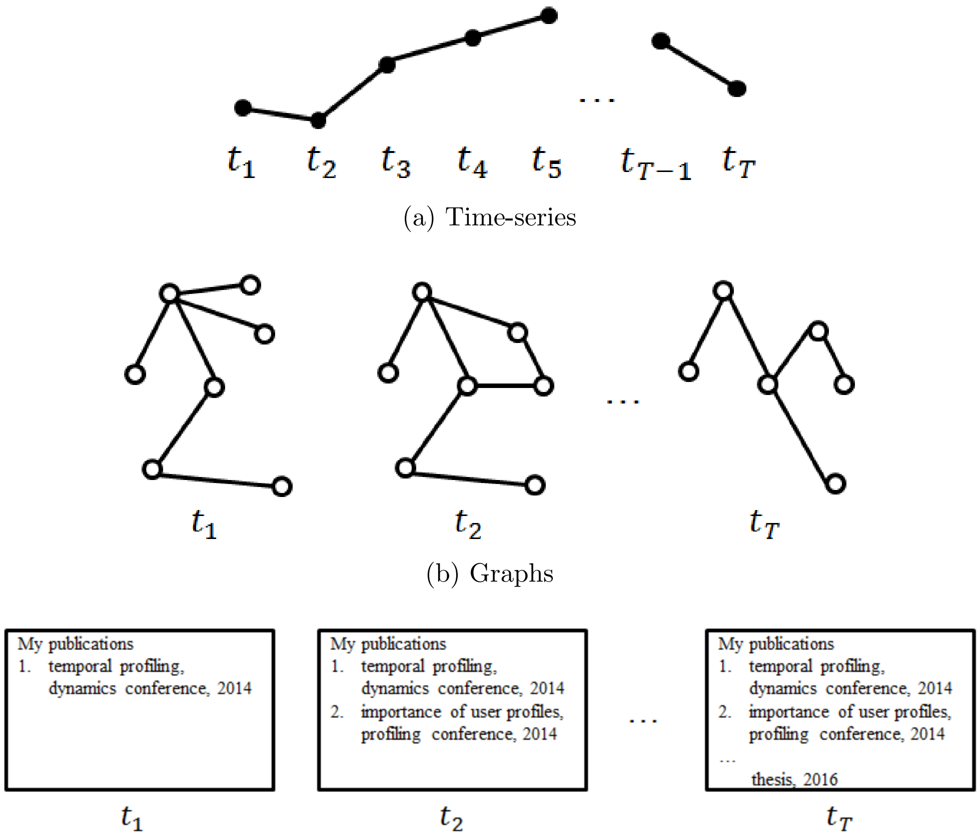
**Profiling methods considering data dynamics** The works [DDD+10; ZI13] have demonstrated that user interests are dynamic. Motivated by these results, several profiling methods [OBP12; SWL+13; MDS01] take data dynamics into consideration. These studies integrate a temporal decay function into their profiling methods. A temporal decay function enables profiling methods to assign term weights and entity weights, considering how recent the information is. Orlandi et al. [OBP12] used the exponential function with different parameters to construct user profiles based on both Twitter and Facebook. Sugiyama and Kan [SK10] also employed the exponential function to build user profiles based on users' scientific publications. Moreover, also using the exponential function, De Francisci Morales et al. [DGL12] developed a recommender system for news articles based on a user's and his neighbors' social media items. On the other hand, Shen et al. [SWL+13] used the sliding window function, which takes only a fixed number of the latest microblog postings into consideration. Finally, Middleton et al. [MDS01] employed inverse time weighting, where original weights are divided by the number of days. However, these temporal decay functions have not been experimentally compared yet.

## 2.2 Profiling the Data Dynamics of Knowledge Graphs

This section first describes different profiling methods to capture and understand data dynamics. Thereafter, it introduces existing works that analyze data dynamics of knowledge graphs.

### 2.2.1 Profiling Data Dynamics

There are various profiling methods to capture and understand data dynamics. In this section, we introduce different profiling methods with respect to three target data types: time-series, graphs, and documents. These data types are summarized in Figure 2.2. We first discuss works whose target is time-series. Time-series, as described in Figure 2.2 (a), refers to data represented as a sequence of numerical values indexed in chronological order. Second, we review works that profile snapshots of a graph, as shown in Figure 2.2 (b). Third, we examine works that analyze snapshots of a document, as depicted in Figure 2.2 (c).



My publications
1. temporal profiling, dynamics conference, 2014

My publications
1. temporal profiling, dynamics conference, 2014
2. importance of user profiles, profiling conference, 2014

My publications
1. temporal profiling, dynamics conference, 2014
2. importance of user profiles, profiling conference, 2014
…
thesis, 2016

(a) Time-series

(b) Graphs

(c) Documents

Figure 2.2: Target data types for profiling data dynamics.

**Time-series** Time-series clustering identifies representative patterns of data dynamics in an unsupervised way. Specifically, it partitions different time-series into a given number of groups based on distance. So far, many different methods of time-series clustering have been developed. According to the results of an extensive experiment of time-series clustering by Wang et al. [WMD+13], the choice of clustering algorithm is less important than the choice of distance measure. Therefore, many researchers have developed distance measures for time-series. The most straightforward distance measure is the Euclidean distance [FRM94]. However, Berndt and Clifford [BC94] also introduced Dynamic Time Warping (DTW), which is used in the speech recognition community, to the data mining community. The DTW allows a time-series to be "stretched" or "compressed" to provide a better match with another time-series. Moreover, several lower-bounding methods have been developed to further accelerate the computation of distance using DTW [Keo02]. Rakthanmanon et al. [RCM+12] introduced four novel ideas to achieve this, and revealed that DTW can exactly search a time-series more quickly than the Euclidean distance can. Furthermore, Batista et al. [BKT+14] presented a complexity-invariant distance measure and showed that it generally produces significant improvements in clustering and classification.

Paparrizos and Gravano proposed k-Shape [PG15], which employs a normalized version of the cross-correlation measure as a distance measure to consider the shapes of time-series. Based on properties of their distance measure, the authors further introduced a novel method to compute cluster centroids. To demonstrate the robustness of the k-Shape, they tested their novel clustering method against partitional, hierarchical, and spectral clustering methods. The results indicated that k-Shape outperformed them all. In another study, Yang and Leskovec [YL11] developed K-Spectral Centroid (K-SC) for the analysis of temporal patterns of social media items. The K-SC is motivated by the fact that the popularity of social media items typically has a steep rise and fall over time [CS08]. Accordingly, it employs a novel distance measure, which is suited to time-series with a steep rise and fall. Their analysis revealed six temporal patterns of attention of online content in microblog postings, blogs, and news articles.

For more details on profiling time-series, we refer to the survey by Esling and Agon [EA12].

**Graphs** The analysis of the data dynamics of graphs is relevant to many different domains, including social networks and biological networks, e .g., protein-protein interactions [AS14]. Leskovec et al. [LKF05] investigated how several real graphs evolve over time, and found that graphs have common data dynamics, such as shrinking diameters and densification. This indicates that graphs are totally different from randomly evolving graphs. In addition, Leskovec et al. [LBK+08] investigated from which nodes and to which nodes a new edge appeared. They observed that new edges were more likely to connect to neighbor nodes, such as a friend of his friend. For more details on analyzing data dynamic of graphs, we refer to the survey by Aggarwal and Subbian [AS14].

Link prediction is also a popular task for graphs, especially for social networks. Sarkar et al. [SCJ12] proposed a nonparametric link prediction method using snapshots of a graph over time. Beyond link prediction, Farajtabar [FWR+15] proposed a temporal point process model called COEVOLVE; it simultaneously models information diffusion on graphs and link generation, since information diffusion and link generations influence each other.

**Documents** Several studies have attempted to predict document changes. This prediction is especially helpful in the context of crawling strategies that download the documents from the web and update their local copies, for instance for indexing and archival. Different features have been used to predict document changes. Cho and Garcia-Molina [CG00; CG03] revealed that probability of changes in documents can be modeled as Poisson process. However, Grimes and O'Brien [GO08] rejected this finding. Santos et al. [SZA+13; SCA+15] used Genetic Programming to generate score functions that produced accurate rankings of documents regarding their probabilities of change. The experiment showed that the number of times that a document was updated in past visits and how much time had passed since the last visit had a large influence on the probabilities. The above works converted snapshots of a document into a time-series by calculating how much it was modified between two successive snapshots and profiled different time-series. Therefore, the studies ignored the contents of documents. In contrast, the following works profile the data dynamics of documents using their contents. Tan and Mitra [TM10] developed a clustering-based incremental crawling strategy that exploits the content of web pages. Thus, it does not need to gather a long history of the web pages before it starts crawling. Their strategy first clusters web pages based on features that correlate to their change frequencies. At each point in time, it crawls a few web pages in each cluster. The web pages in the cluster are then all downloaded and updated only if the crawled web pages of that cluster have many changes. In terms of the features for clustering, the strategy uses static features such as content (e .g., terms in documents, the number of images), URL (e.g., name of the top-level domain), and linkage features (e.g., the number of incoming links). In addition, it exploits dynamic features that calculate how much each content feature and linkage feature have changed in the two latest successive snapshots. Tan and Mitra's experiment revealed that the combined features (i.e., both content features and dynamic features) were best for the crawling strategies. Finally, Radinsky and Bennett [RB13] reported that content of documents or neighbor documents significantly improved the prediction of probability of changes.

### 2.2.2 Data Dynamics of Knowledge Graphs

Since knowledge graphs are maintained by humans, they are subject to changes. Therefore, several works have investigated the data dynamics of knowledge graphs. In this section, we introduce works that capture the data dynamics of knowledge graphs. Thereafter, we describe several works that aim at maintaining the integrity of knowledge graphs.

Please note that this thesis sees the Linked Open Data (LOD) cloud as a collection of knowledge graphs, and an RDF document as a web document that provides data of a knowledge graph.

**Profiling the data dynamics of knowledge graphs** Umbrich et al. [UKL10] investigated the data dynamics in the LOD cloud, focusing on entities. They defined and represented entities as a set of triples that share a common subject URI. To group entities with similar data dynamics, they applied k-means clustering. Their manual inspection revealed that entities from the same pay-level-domains (PLDs) were often found in the same clusters. However, they only considered whether there was a change or not, and not the amount of change of the entities, i.e., the number of triples that changed in entities. In addition, Umbrich et al. [UHH+10] investigated the data dynamics of the LOD cloud, focusing on entities and RDF documents. As a dataset, they collected weekly snapshots of the neighbors of the Tim Berners-Lee FOAF file[11] for 24 weeks. They observed that half of the RDF documents that changed had a change frequency of more than 3 months. In contrast, half of the entities had a change frequency of less than a week. In addition, the authors could not verify that the change frequency of the RDF documents and entities followed a Poisson process, as was observed in web documents [CG03]. In another study, Popitsch and Haslhofer [PH11] provided statistics about changes of entities between two successive DBpedia

[11] `https://www.w3.org/People/Berners-Lee/card.rdf`, last accessed on 08/30/2017

snapshots with respect to four OWL classes (i.e., `person`, `organization`, `place`, and `work`). Their results suggested that DBpedia grew continuously. In terms of OWL classes, entities belonging to the `person` class were active, as many entities were removed and created. However, the focus of their work was not to analyze the data dynamics of entities, but to develop an effective entity change detection framework to avoid broken links. Therefore, they did not conduct a fine-grained analysis.

In 2012, Käfer et al. [KUH+12] launched the Dynamic Linked Data Observatory (DyLDO). The DyLDO collected weekly snapshots of 86,696 RDF documents on the LOD cloud for four years, until it recently stopped providing updates. The snapshots contain both well-known data sources, such as DBpedia and Freebase, and lesser-known ones. Since the DyLDO made the snapshots publicly available, it encouraged many researchers to study the data dynamics on the LOD cloud. Käfer et al. [KAU+13] conducted an analysis based on 29 weekly snapshots of the DyLDO. They found that 5.0% of RDF documents had gone offline, and 62.2% of them had no change. In addition, they conducted an analysis focusing on triples. The result indicated that the additions of triples were much more frequent than deletions. Furthermore, the authors observed that object literals were the most dynamic elements of triples. In contrast, predicates (i.e., properties) and RDF types defined by the predicate `rdf:type` were static. They identified that the most dynamic predicates were often about trivial time stamps. In another study, Gottron and Gottron [GG14] developed different index models for RDF documents, and evaluated the accuracy of these models over time with regard to finding relevant RDF documents. They used a DyLDO dataset of 80 weeks in their experiment, which revealed an increasing divergence of the index due to the data dynamics of the RDF documents. However, index models based on schema information seemed to be relatively stable. Moreover, Dividino et al. [DKG14] measured how often the `last-modified` field in the HTTP header of RDF documents was available and how often it was correctly used. Using the DyLDO dataset, their analysis revealed that on average only 15% of the RDF documents provided some value for the `last-modified` field, and in turn only 52% provided accurate value. Therefore, it is not practical to study the data dynamics on the LOD cloud using the `last-modified` field. Instead, to represent the data dynamics of RDF documents, Dividino et al. [DGS+14] proposed a monotone, nonnegative function that returns a single numerical value. Using this function, Dividino et al. [DGS15] developed a novel crawling strategy to keep local copies of RDF documents up-to-date while respecting limited bandwidth. The experiment revealed that crawling strategies based on the novel function performed best when compared to those based on the RDF documents' age, PageRank, or size. This result suggests that the novel function can correctly represent the data dynamics of RDF documents and enable the prediction of their future changes.

**Keeping the integrity of knowledge graphs** It is difficult to keep the information in knowledge graphs up-to-date, since the real world continuously changes over time. Therefore, knowledge graphs need to be updated. In practice, many editors make changes on knowledge graphs such as Wikidata [VK14]. Changes are represented as additions or deletions of triples. While the majority of changes are correct, knowledge graphs also receive incorrect changes due to vandalism, carelessness, and misunderstanding by editors. Thus, the change verification for knowledge graphs is demanding to keep the integrity of knowledge graphs. Change verification automatically judges incoming changes as correct or incorrect one.

So far, only a few studies have investigated methods of change verification. Tan et al. [TAI+14] proposed a method using three categories of features to automatically classify changes made for Freebase into correct or incorrect ones. The features are categorized into triple feature, editor history, and editor expertise. In terms of triple feature, the method uses only the predicate, but it demonstrates the highest effectiveness. Editor history includes numbers of correct and incorrect changes made by the editor in the past, as well as the age of his account. Editor expertise refers to how well editors make changes in each domain, such as sports and science. Tan et al.'s experiment demonstrated that the classifier using all features performed best, and that the triple feature is the most effective feature. In terms of classification algorithms, logistic regression outperforms Gradboost [DS09] as well as Perceptron [FS99] in their experiment. Later, Heindorf et al. [HPS+16] proposed a set of 47 features to verify changes made to Wikidata. Their features can be categorized into two groups: content features and context features. Content features include textual features, triple features, and comment features, whereas context features contain editor features (e .g., an editor's experience, country), entity features (e .g., their popularity), and revision features. In their experiment, classifiers based on all features showed the highest performance. The authors observed that content features and context features contributed to improving precision and recall, respectively. Classification algorithms using the random forest [Bre01] in combination with multiple-instance learning obtained the best performance. While these features performed well, however, some of them can be applied only to Wikidata, such as comment features.

Apart from change verification, different methods of knowledge graph refinement have been studied [Pau16]. The goal of knowledge graph refinement is to add missing triples (i .e., completion) or identify erroneous ones (i .e., error detection) in a static knowledge graph [Pau16]. Thus, refinement methods are relevant for change verification. Nickel et al. [NTK12] used matrix factorization to predict entity types in YAGO. Socher et al. [SCM+13] predicted the existence of a relation between two entities by training a tensor neural network based on chains of other relations. Dong et al. [DGH+14] employed the path ranking algorithm [LMC11] and the neural network model [SCM+13] to judge whether an extracted new triple should be added to a knowledge graph. Regarding error detection in knowledge graphs, reasoning determines whether a given set of triples is free of contradictions or not [LLB+09]. However, this requires a rich ontology. In this vein, Guéret et al. [GGS+12] used topological features such as degree, clustering coefficient, and centrality to define metrics for detecting wrong triples in knowledge graphs. They compared the true distributions of those metrics to the ones that were ideally expected, e .g., a power law distribution for the degree of entities. Then, they marked links that deviated from the ideal distributions as suspicious. However, the above methods are difficult to apply to verify incoming changes online.

## 2.3 Summary

In this section, we summarize the literature that we introduced so far. In terms of profiling users, we first introduced term-based profiling methods as well as several profiling methods using knowledge graphs. The term-based profiling methods have statistical strength, but do not work for short documents. On the other hand, the profiling methods using knowledge graphs can cover this drawback by using semantics of knowledge graphs. In the subsequent chapter, we present a novel profiling method that combines the statistical strength of CF-IDF and the semantics originating from a knowledge graph. We also reviewed works of profiling methods for data dynamics. We categorized these methods by target data types: methods for time-series, those for snapshots of a graph, and those for snapshots of a document. Then, we have shown works that analyzed the data dynamics of knowledge graphs. In general, these works focused on the amount of changes in RDF documents between two successive snapshots or during an entire observation period. Thus, they do not look into the changes over a larger period of time and what influences on the data dynamics of knowledge graphs. In Chapter 6, we investigate how the content and structure (i .e., topology) of knowledge graphs influence on the data dynamics of knowledge graphs.

# Chapter 3

# Foundational Definitions

Profiling methods play an important role in different tasks such as information retrieval [TDH05; Das98; YLH+03; MDL+00; Mob07] and recommender systems [KJN08; FEB+02; KB06]. In particular, user profiling is indispensable to overcome the information overload problem. Since many users publish their interests and thoughts on social media platforms such as Twitter and Facebook, these published social media items are a promising source for profiling users. However, these items are typically short. Therefore, it is highly difficult to profile users using traditional term-based profiling methods such as TF-IDF and topic modeling without external knowledge [HD10].

In this chapter, we first introduce foundational definitions for profiling documents and users in Section 3.1. Thereafter, Section 3.2 describes different profiling methods using knowledge graphs. In the last decade, many different knowledge graphs such as DBpedia [ABK+07] and Wikidata [VK14] have been developed with the objective of encouraging information reuse and discovery. Therefore, we assume that knowledge graphs can assist in profiling methods. Profiling methods using knowledge graphs represent a document as set of entities and their weights. In Section 3.3, we provide several temporal decay functions used by profiling methods to take the data dynamics of user interests into account. The data dynamics is important for profiling users, because user interests change over time. Temporal decay functions model the data dynamics of user interests by assigning a larger weight to newer social media items, as older social media items become stale and may not reflect users' current interests.

## 3.1 Foundational Definitions for Profiling Documents and Users

We first introduce the formalization of profiling documents. The formalization is applicable to both term-based profiling methods and profiling methods using knowledge graphs.

**Definition 3.1** (Document Profile)**.** *Formally, a document profile is a vector of term weights (or entity weights). Let $d$ be a document and $A = \{a_1, a_2, \ldots, a_{|A|}\}$ be a set of unique terms in a document collection (or a set of entities in a knowledge graph). A profiling function $\Phi$ produces a profile for a given document $d$ as:*

$$\Phi : d \rightarrow \{w(a_1, d), w(a_2, d), \ldots, w(a_{|A|}, d)\},$$

*where $w(a, d)$ is a weight of the term (or the entity) $a$ for the document $d$.*

We subsequently define how to profile a user based on his social media items such as microblog postings. Below, we introduce how to profile a user.

**Definition 3.2** (User Profile)**.** *Let $J_u = \{j_1, j_2, \ldots, j_{|J_u|}\}$ be a set of social media items produced by a user $u$. A profiling function $\Phi$ produces a user profile for a given set of social media items $J_u$ as:*

$$\Phi : J_u \rightarrow \{w(a_1, J_u), w(a_2, J_u), \ldots, w(a_{|A|}, J_u)\},$$

*where $w(a, J_u)$ is a weight of the term (or the entity) $a$ for a set of social media items $J_u$. $w(a, J_u)$ is computed as:*

$$w(a, J_u) = \sum_{j \in J_u} w(a, j).$$

In terms of the source of user profiles, Chen et al. [CNN+10] developed a recommender system that suggests URLs based on a user's own microblog postings or those of her followees. The experiment revealed that user profiles based on the user's own microblog postings made better recommendations than those based on those of the followees. Therefore, we build up user profiles based on social media items produced by the users themselves.

For profiling documents and users, the weighting function $w$ is essential. This function is defined below.

**Definition 3.3** (Weighting Function). *$w(a, d)$ and $w(a, j)$ are a weight of the term (or the entity) $a$ for a document $d$ and for a social media item $j$, respectively. They are computed as:*

$$w(a, d) = \nu(a, d) \cdot \mu(t(d)).$$
$$w(a, j) = \nu(a, j) \cdot \mu(t(j)).$$

*$\nu$ is a term (or entity) relevance function. $\nu(a, d)$ and $\nu(a, j)$ represent the degree of association of the term (or entity) $a$ with the document $d$ and the social media item $j$, respectively. $\mu$ is a temporal decay function. $\mu(t)$ returns the weight with regard to a given time $t$. $t(d)$ and $t(j)$ refer to the time stamp of the document $d$ and the social media item $j$, respectively.*

In Section 3.2, we introduce different term-based profiling methods and profiling methods using knowledge graphs. We especially focus on the term (or entity) relevance function $\nu$. Section 3.3 then presents different temporal decay functions $\mu$, to take data dynamics into account.

## 3.2 Profiling Methods with Knowledge Graphs

We first introduce traditional term-based profiling methods used in this thesis in Section 3.2.1. Then, Section 3.2.2 provides different profiling methods using knowledge graphs. We focus on the term (or entity) relevance function $\nu$ in this section.

### 3.2.1 Term-based Profiling Methods

We start from the formalization of term-based profiling methods. We refer to $V = \{v_1, v_2, \ldots, v_{|V|}\}$ as the dictionary, which is a set of unique terms in a document collection. A set of unique terms $V$ is computed by applying natural language processing, such as tokenization, stop word removal, and lemmatization. The formalizations of profiling documents and users using term-based profiling methods are described as replacing $a$ with $v$ and $A$ with $V$ in Definitions 3.1 and 3.2, respectively. In addition, the weighting function for term-based profiling methods is also given by replacing $a$ with $v$ in Definition 3.3.

Below, we introduce the best-known term-based profiling methods, focusing on the term relevance function $\nu$. Please note that $\nu(v, j)$ is computed in the same way as $\nu(v, d)$, because both documents and social media items consist of terms.

**Term Frequency-Inverse Document Frequency (TF-IDF)** TF-IDF [SB-88; SM86; SWY75] is the best-known profiling method that was originally introduced for information retrieval. It is still a robust baseline in information retrieval and text mining, although it was first developed more than 40 years ago [SWY75]. TF counts the frequency with which a term appears in a document. This is based on the assumption that multiple appearances of a term in a document are more relevant than single appearances. On the other hand, IDF is based on the assumption that rare terms are more relevant than frequent terms. In other words, terms that occur frequently in one document but rarely in the rest of the document corpus are more likely to be relevant to that document. TF-IDF computes a term weight as:

$$\nu_{tfidf}(v,d) = \frac{freq(v,d)}{\sum_{v_i \in V} freq(v_i,d)} \cdot \log \frac{|D|}{|d \in D : v \in d|}. \tag{3.1}$$

$freq(v,d)$ returns the frequency of a term $v$ in a document $d$. Thus, $\sum_{v_i \in V} freq(v_i,d)$ denotes the total number of terms in a document $d$, which is equal to the length of a document $d$. $|D|$ denotes the number of documents in a document corpus. $|d \in D : v \in d|$ counts the number of documents that contain a term $v$ in the document corpus $D$.

**Okapi BM25 (BM25)** BM25 [RWJ+94; RW94] is the state of the art for nearly 20 years. It computes a term weight as:

$$\nu_{bm25}(v,d) = \frac{freq(v,d) \cdot (\alpha+1)}{freq(v,d) + \alpha \cdot (1 - \beta + \beta \cdot \frac{\sum_{v_i \in V} freq(v_i,d)}{avgdl})} \cdot idfbm25(v,d). \tag{3.2}$$

*avgdl* denotes the average length (i .e., the average total number of terms) of documents in the document corpus. Both $\alpha$ and $\beta$ are parameters. $\alpha$ is a positive tuning parameter that calibrates the scaling of the term frequency. If $\alpha = 0$, the model is interpreted as a binary model (i .e., whether a term appears in a document). A large value of $\alpha$ corresponds to using raw term frequency. $\beta$ is another parameter, and is $\beta \in [0,1]$. It determines the scaling by document length. $\beta = 1$ corresponds to fully scaling the term weight by the document length, whereas $\beta = 0$ indicates no normalization by the document length. For these parameters, Manning et al. [MRS08] suggested that $\alpha \in [1.2, 2.0]$ and $\beta = 0.75$ are the best general settings.

$idfbm25(v, d)$ is defined as:

$$idfbm25(v, d) = \log \frac{|D| - |d \in D : v \in d| + 0.5}{|d \in D : v \in d| + 0.5}. \tag{3.3}$$

Please note that if a term occurs in over half of the documents in the document corpus, then $idfbm25(v, d)$ provides a negative value, which is presumably undesirable. However, assuming the use of stop word removal, this usually does not happen [MRS08].

TF-IDF and BM25 have been robust baselines for decades. Especially TF-IDF has been widely used, since it requires no parameter.

### 3.2.2 Profiling Methods Using Knowledge Graphs

In this section, we first define a knowledge graph. Subsequently, we describe how to detect entities from documents. Finally, we introduce different profiling methods using knowledge graphs.

**Definition 3.4** (Knowledge Graph, Triple, Entity)**.** *According to the standard RDF-based knowledge graph, a triple $\langle s, p, o \rangle$ consists of subject $s$, predicate $p$, and object $o$. Let $R$ and $L$ be the respective sets of all URIs and literals. A URI $r \in R$ refers to an entity or a predicate (i .e, property in RDF). A literal $l \in L$ provides a value such as a label and a number. In a triple $\langle s, p, o \rangle$, a subject $s \in R$ is a URI, a predicate $p \in R$ is a URI, and an object $o \in R \cup L$ is a URI or a literal. Naturally, a knowledge graph can be seen as a directed graph:*

$$G = (R \cup L, R \times R \times (R \cup L)).$$

*A node is a URI or a literal that is used as subject or object. The set of edges $E$ is considered as triples:*

$$E = R \times R \times (R \cup L).$$

*A URI $r$ is an entity if it satisfies the condition $\exists \langle r, \cdot, \cdot \rangle \in E \vee \langle \cdot, \cdot, r \rangle \in E$. Let $Q$ be a set of all entities that satisfy the above condition, which are the subset of all URIs, thus $Q \subset R$.*

Although a subject and object can be a blank node[1], we ignore it in this thesis as the use of blank nodes is discouraged for Linked Data [BHB09].

[1] `https://www.w3.org/TR/n-triples/#BNodes`, last accessed on 08/30/2017

Figure 3.1 shows a small example of a knowledge graph. Please note that dbr, dbp-owl, and rdfs are namespace prefixes that are originally http://dbpedia.org/resource/, http://dbpedia.org/ontology/, and http://www.w3.org/1999/02/22-rdf-syntax-ns#. These namespace prefixes are defined in a knowledge graph. In the example, there are four URIs (i.e., dbr:Kiel, dbr:Ulf_Kaempfer, dbp-owl:mayor, rdfs:label) and one literal (i.e., Kiel). Among the four URIs, we consider dbr:Kiel and dbr:Ulf_Kaempfer as entities, since they are used as a subject or object. The example delivers two facts (i.e., triples): first, that the mayor of Kiel is Ulf Kaempfer, and second, that the label of the entity Kiel is "Kiel."

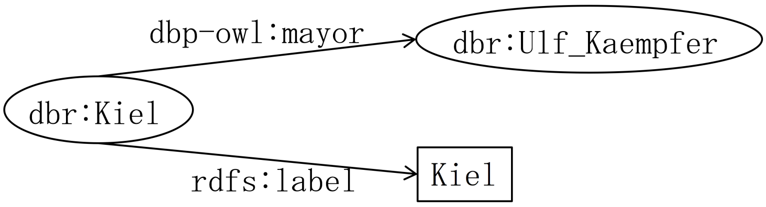


Figure 3.1: An example of a knowledge graph.

Next, we describe how to detect entities from documents. We assume that labels of entities are available in a knowledge graph. These labels are given by a triple such as <http://dbpedia.org/resource/Apple><http://www.w3.org/2000/01/rdf-schema\#label>"Apfel"@de.. In this triple, the subject URI <http://dbpedia.org/resource/Apple> indicates the entity "apple." The predicate URI <http://www.w3.org/2000/01/rdf-schema\#label> is a property that defines a label. Please note there are other properties that define a label. For example, Linked Open Vocabularies (LOV) [VAP+17], a catalog of reusable vocabularies, shows different terms[2] to define labels. In addition, we may use the properties that provide synonyms of entities, such as http://dbpedia.org/ontology/synonym, and abbreviations of entities such as http://dbpedia.org/ontology/abbreviation. In terms of the object "Apfel"@de, "Apfel" is a label and @de indicates a language of the label (in this case, German). Using the labels, we extract entities by a naive string matching method. To reduce the number of false positives, the method usually takes labels that consist of at least several characters. In addition, we assume that entities do not share labels. As alternatives to the naive string matching method, profiling methods may use more sophisticated entity extraction methods [GGL+16; FDK16; TCL+16; PMA+16]. In addition, several tools for entity extraction have

[2] http://lov.okfn.org/dataset/lov/terms?q=label, last accessed on 08/30/2017

been deployed such as DBpediaSpotLight [MJG+11], Alchemy[3], and OpenCalais[4]. Rizzo and Troncy [RT11] conducted an extensive evaluation of these tools.

The formalizations of profiling documents and users with a knowledge graph are described by replacing $a$ with $r$ and $A$ with $Q$ in Definitions 3.1 and 3.2, respectively. In addition, the weighting function of profiling methods using knowledge graphs is also given by replacing $a$ with $r$ in Definition 3.3.

**Profiling method using a list of domain-specific entities** We first introduce profiling methods using a list of entities. Specifically, we focus on the entity relevance function $\nu$.

**Frequency (Freq)** The frequency (i .e., number of appearances) of an entity is provided as a weight.

$$\nu_{freq}(r, d) = freq(r, d). \tag{3.4}$$

$freq(r, d)$ returns the number of appearances of an entity $r$ in a document $d$. This profiling method was used by Abel et al. [AHK11] to extract users' professional interests.

**Concept Frequency-Inverse Document Frequency (CF-IDF)** CF-IDF [GIF+11] is an extension of the traditional TF-IDF [SB88; SM86; SWY75] that counts entities instead of terms. CF-IDF gives a weight as follows:

$$\nu_{cfidf}(r, d) = \frac{freq(r, d)}{\sum_{r_i \in Q} freq(r_i, d)} \cdot \log \frac{|D|}{|d \in D : r \in d|}. \tag{3.5}$$

$\sum_{r_i \in Q} freq(r_i, d)$ represents the total number of entities in a document d (i .e., the length of a document $d$). $|D|$ denotes the number of documents in a document corpus. $|d \in D : r \in d|$ indicates the number of documents that contain an entity $r$ in the document corpus $D$.

**BM25C** BM25C [NGS15] is a novel profiling method that is an extension of BM25 [RWB99]. Like CF-IDF, it counts entities instead of terms and uses them in BM25.

$$\nu_{bm25c}(r, d) = \frac{freq(r, d) \cdot (\alpha + 1)}{freq(r, d) + \alpha \cdot (1 - \beta + \beta \cdot \frac{\sum_{r_i \in Q} freq(r_i, d)}{avgdl})} \cdot bm25cidf(r, d), \tag{3.6}$$

[3] `https://www.ibm.com/watson/developercloud/alchemy-language.html`, last accessed on 08/30/2017

[4] `http://www.opencalais.com/`, last accessed on 08/30/2017

where *avgdl* denotes the average length (i .e., the average number of entities) of documents in the document corpus. Both $\alpha$ and $\beta$ are parameters. $bm25cidf(r, D)$ is defined as:

$$bm25cidf(r, d) = \log \frac{|D| - |d \in D : r \in d| + 0.5}{|d \in D : r \in d| + 0.5}. \tag{3.7}$$

**Profiling method using hierarchical knowledge graphs** The above profiling methods exploit the labels encoded in a knowledge graph. However, they do not leverage its structure. Below, we introduce profiling methods that do take into account the structure. Specifically, this thesis utilizes the hierarchical structure of a knowledge graph, because there are a lot of hierarchical knowledge graphs in different domains. For example, taxonomies and thesauri used in libraries and classification systems [YKS+14] have the hierarchical structure. These graphs are usually defined following the Simple Knowledge Organization System (SKOS) specifications[5]. The SKOS is a W3C recommendation designed to represent thesauri, taxonomies, or any other type of structured controlled vocabulary. Following the SKOS specifications, many different hierarchical knowledge graphs have been published in different domains. The list of freely available hierarchical knowledge graphs[6] is maintained by the W3C. In addition, these knowledge graphs are often crafted manually by domain experts, and are thus of high quality.

The SKOS includes two basic categories of semantic relations[7]: hierarchical and associative. Hierarchical relations include broader and narrower relations indicating that one is in some way more general (i .e., broader) than the other (i .e., narrower). An associative relation between two entities indicates that they are related. In this thesis, we exploit only hierarchical relations.

Below, we define a hierarchical knowledge graph.

**Definition 3.5** (Hierarchical Knowledge Graph). *A knowledge graph is a hierarchical knowledge graph if the following criteria are satisfied:*

1. *It is a directed acyclic graph (DAG).*

2. *Entities $r \in Q$ are connected by a relation, either* `broader` *or* `narrower`*.*

A hierarchical knowledge graph can be seen as a DAG, which is a graph that contains no cycle. The definition allows poly-hierarchical re-

[5] `https://www.w3.org/2004/02/skos/`, last accessed on 08/30/2017

[6] `http://www.w3.org/2001/sw/wiki/SKOS/Datasets`, last accessed on 08/30/2017

[7] `https://www.w3.org/TR/skos-reference/#semantic-relations`, last accessed on 30/08/2017

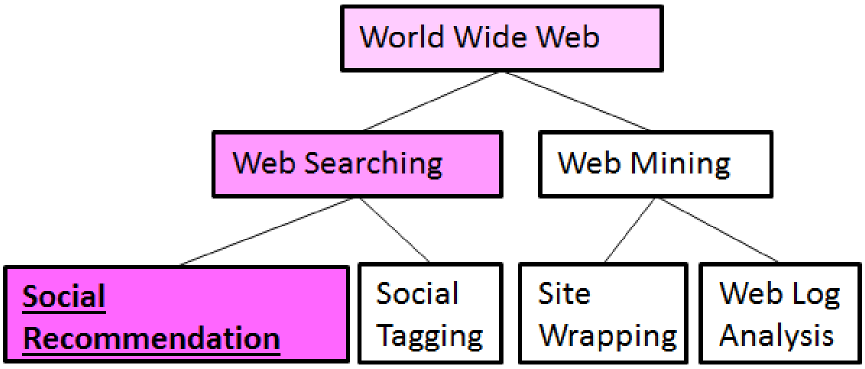


Figure 3.2: An example of a hierarchical knowledge graph.

lations. In other words, an entity may have more than one broader entity. In the SKOS specifications, broader and narrower relations are defined by the properties `http://www.w3.org/2004/02/skos/core\#broader` and `http://www.w3.org/2004/02/skos/core\#narrower`, respectively. For example, the statement that "apple" is narrower than "fruit" is represented as `<http://dbpedia.org/resource/Fruit><http://www.w3.org/2004/02/skos/core\#narrower><http://dbpedia.org/resource/Apple>.`. In addition, the SKOS specifications define labels by the properties `http://www.w3.org/2004/02/skos/core\#prefLabel` and `http://www.w3.org/2004/02/skos/core\#altLabel`. `prefLabel` provides a preferred label (i .e., main label) for an entity. An entity has at most one `prefLabel`. `altLabel` provides labels other than the preferred label. Figure 3.2 shows an example of a hierarchical knowledge graph. In the example, the broader entity of the entity "web mining" is the entity "world wide web," while the narrower entities of the entity "web mining" are the entities "site wrapping" and "web log analysis."

Based on the definition of a hierarchical knowledge graph, we describe different profiling methods that exploit the hierarchical structure. These profiling methods have the advantage that they can extract entities that are not mentioned directly but are nevertheless relevant to a document. Specifically, they boost the weights of broader entities. Using the example in Figure 3.2, the entities "web searching" and "world wide web" are activated and obtain non-zero weights, even if a document contains only the entity "social recommendation." Thus, it is expected that these profiling methods are useful especially for short documents such as social media items. They can extract sufficient amount of entities from short documents by boosting the weights of relevant entities. Below, we describe profiling methods using the hierarchical structure of knowledge graphs.

**Basic Spreading Activation (Basic)** The basic spreading activation [KJV+14] uses the spreading activation [CL75], which is a propagation function.

$$\nu_{basic}(r,d) = freq(r,d) + \lambda \cdot \sum_{r_i \in LO(r)} \nu_{basic}(r_i,d). \tag{3.8}$$

$LO(r)$ returns a set of entities located in a lower order of the entity $r$ in the hierarchical knowledge graph $G$. Using the example in Figure 3.2, $LO($"world wide web"$)$ returns the entities "web searching" and "web mining". $\lambda$ is a decay parameter. Kapanipathi et al. [KJV+14] used this method in their study to extract user interests from microblog postings.

**Bell Spreading Activation (Bell)** Kapanipathi et al. [KJV+14] observed that the distribution of entities across the different levels of a hierarchical knowledge graph follows a bell curve. Based on this observation, they developed the bell spreading activation as defined in Equation 3.9.

$$\nu_{bell}(r,d) = freq(r,d) + \frac{1}{|LO(r)|} \cdot \sum_{r_i \in LO(r)} \nu_{bell}(r_i,d). \tag{3.9}$$

**Bell Logarithmic Spreading Activation (BellLog)** Kapanipathi et al. [KJV+14] introduced the logarithmic scale for the bell spreading activation to reduce the impact of the raw count of the number of entities.

$$\nu_{belllog}(r,d) = freq(r,d) + \frac{1}{\log_{10}|LO(r)|} \cdot \sum_{r_i \in LO(r)} \nu_{belllog}(r_i,d). \tag{3.10}$$

Below, we provide novel entity weighting functions that we have developed. These functions make use of both the statistical strength of CF-IDF and the semantics from the structure of a knowledge graph.

**Hierarchical Concept Frequency-Inverse Document Frequency (HCF-IDF)** HCF-IDF [NGS15] is a novel profiling method that is an extension of CF-IDF and leverages the hierarchical structure of a knowledge graph. Thus, HCF-IDF benefits from both the statistical strength of CF-IDF and the semantics from a knowledge graph. HCF-IDF computes an entity weight as follows:

$$\nu_{hcfidf}(r,d) = \nu_{belllog}(r,d) \cdot \log \frac{|D|}{|d \in D : r \in d|}. \tag{3.11}$$

$|d \in D : r \in d|$ denotes the number of documents containing an entity $r$ after applying BellLog. As spreading activation, HCF-IDF exploits BellLog, since according to Kapanipathi et al. [KJV+14], BellLog performs best (except for the method *PriorityInterest*, which is not applicable here).

**BM25HC** BM25HC [NGS15] is a novel profiling method that is an extension of BM25C. It uses the hierarchical structure of a knowledge graph.

$$\nu_{bm25hc}(r,d) = \frac{\nu_{belllog}(r,d) \cdot (\alpha + 1)}{\nu_{belllog}(r,d) + \alpha \cdot (1 - \beta + \beta \cdot \frac{\sum_{r_i \in Q} freq(r_i,d)}{avgdl})} \cdot bm25cidf(r,d). \tag{3.12}$$

*avgdl* denotes the average length (i .e., the average number of entities) of documents in a document corpus after applying BellLog. $bm25cidf(r,d)$ is shown in Equation 3.7. Again, $|d \in D : r \in d|$ denotes the number of documents containing an entity $r$ after applying BellLog in BM25HC.

## 3.3 Temporal Decay Functions for Profiling Methods

Documents and social media items have different time stamps. Since older information is intuitively less important, we should take into account these time stamps. In most cases, interests that have only been expressed by a user in the past are less relevant than interests that have been expressed recently. Therefore, we can state that user interests decay with the time [Orl14]. This decay can be modeled by a temporal decay function $\mu$, which takes a point in time $t$ (i .e., time stamp) as an argument. $\mu$ is used to compute a weight of the term (or the entity) as described in Definition 3.3. So far, different temporal decay functions have been used by profiling methods. Below, we introduce the best-known ones that are used in this thesis.

**No Temporal Decay** The no temporal decay function does not take into account data dynamics. Thus, it gives an equal weight to all items (i .e., documents, social media items).

$$\mu_{nd}(t) = 1. \tag{3.13}$$

**Sliding Window** There are two kinds of sliding window functions. The first is (a) the function whose window size is defined by the number of items [KJN08], and the other is (b) the function whose window size is set by the period of time [SC98]. The function (a) is employed to identify relatively short-term profiles (e .g., user interests based on web browsing histories) [KJN08]. In contrast, the function (b) is used to identify long-term profiles [SC98]. In this thesis, we aim to profile users and analyze the data that is long-term. Thus, we take the function (b). The sliding window function is defined as:

$$\mu_{sw}(t) = \begin{cases} 1 & \text{for } t_{current} - t < t_{window} \\ 0 & \text{for } t_{current} - t \geq t_{window} \end{cases} . \tag{3.14}$$

$t_{current}$ denotes the current point in time. $t_{window}$ is a window size that refers to a period of time (e .g., one month). Thus, the sliding window function only considers items that have been produced within the period of $t_{window}$.

**Exponential** Different psychologists have observed that the forgetting curve [Ebb85; Ebb13] follows the exponential function [AH11; AT97]. Motivated by these observations, many profiling methods have employed this function [DL05; SK10; OBP12; DGL12]. The function is defined as:

$$\mu_{exp}(t) = e^{-(t_{current} - t)/\tau}, \tag{3.15}$$

where $\tau$ is a parameter that controls the speed of forgetting.

# Chapter 4

# Application I: Recommender System for Researchers

The first application is a recommender system for researchers. With the rapid growth of academic communities, different academic social network platforms have been developed such as ResearchGate[1]. Since many researchers are available at the platform, it is difficult for users to find out interesting researchers. This challenge restricts communications among researchers in a traditional way, wherein collaboration is done with only individuals they already know [XGH+12].

The goal of our recommender system is to mitigate this information overload problem by suggesting researchers who might interest users. In the last decade, researchers have also been highly active on social network platforms such as Twitter [LPB+10]. Thus, social media items published by a user are a promising source for building up a user profile, which can be used for recommender systems. However, it is difficult to detect users' professional interests due to their implicit nature. To address this problem, we employ profiling methods using knowledge graphs introduced in Section 3.2, as these profiling methods are assumed to be able to extract implicit user interests. Our recommender system suggests relevant researchers based on a user's social media items. While user profiles are constructed based on their own social media items, our recommender system profiles researchers (i .e., candidate items) based on their own scientific publications. Researchers who score higher similarity with a user are recommended to him.

In our experiment, we examine the recommender system using three factors. The first is *Profiling Method*. In this factor, we compare two term-based profiling methods and eight profiling methods using knowledge graphs described in Section 3.2. The second is *Usage of Older Scientific Publications*, which in-

[1] `https://www.researchgate.net`, last accessed on 08/30/2017

vestigates how older scientific publications used for researcher profiles influence the recommendation performance. Finally, the third factor is *Content Richness*, where we examine the influence of the content richness on the recommendation performance. Specifically, we compare researcher profiles based on only titles of scientific publications and those based on both titles and abstracts. In addition, we analyze the correlation of the recommendation performance with the number of social media items for user profiles and the number of scientific publications for researcher profiles. The experiment employs Twitter as the social media platform due to its popularity in the scientific community [LPB+10]. We conduct the experiment in two domains: computer science and medicine.

In Section 4.1, we formalize the problem tackled in this chapter. Subsequently, Section 4.2 introduces the three experimental factors and their details, and Section 4.3 describes the details of the experiment. Thereafter, the results of the experiment are reported in Section 4.4 and discussed in Section 4.5.

## 4.1 Problem Statement

We tackle with a problem of recommending researchers based on user's social media items. The problem consists of three parts:

**Profiling a user** A user $u$ generates social media items $J_u$. A user profile is represented by $\Phi(J_u)$ as defined in Definition 3.2.

**Profiling researchers** Let $D$ be a set of researchers who are candidate items of the recommender system. Each researcher $d \in D$ is represented as a collection of her scientific publications. We treat the collection of scientific publications authored by a researcher as one *single* scientific publication document $d$. A researcher profile is provided by $\Phi(d)$ as defined in Definition 3.1.

**Ranking researchers to a user** Researchers are ranked based on similarity scores between a user profile $\Phi(J_u)$ and a researcher profile $\Phi(d)$. A function that outputs similarity scores between them is defined as $\sigma : \Phi(J_u), \Phi(d) \rightarrow [0, 1]$. Researchers whose similarity scores are higher are preferentially recommended to the user.

## 4.2 Design of the Experimental Factors

The recommender system is composed of three experimental factors: *Profiling Method*, *Usage of Older Scientific Publications*, and *Content Richness*. We detail the factor *Profiling Method* in Section 4.2.1, *Usage of Older Scientific Publications* in Section 4.2.2, and *Content Richness* in Section 4.2.3. Subsequently, we describe the similarity function $\sigma$ used in our experiments in Section 4.2.4.

### 4.2.1 Profiling Methods

In total, we experiment with two term-based profiling methods and eight profiling methods using knowledge graphs, which are introduced in Section 3.2. We use both term and entity relevance function $\nu$. Please note that this chapter does not investigate the temporal decay function $\mu$. Thus, we apply the no temporal decay function as defined in Equation 3.13 to all user profiles and researcher profiles.

We employ the following two term-based profiling methods.

- **TF-IDF** (c.f., Equation 3.1)
- **BM25** (c.f., Equation 3.2)

We further compare eight profiling methods using knowledge graphs. As entity extraction, we employ the naive string matching approach in all profiling methods. To reduce the number of false positives, we only take labels composed of at least four characters. We employ a domain-specific hierarchical knowledge graph, which can avoid noise such as what Abel et al. [AHK11] observed in user profiles. This is especially beneficial for social media items, since they frequently contain private interests that are irrelevant to professional ones. The eight profiling methods using knowledge graphs are listed below.

- **Freq** (c.f., Equation 3.1)
- **Basic** (c.f., Equation 3.8)
- **Bell** (c.f., Equation 3.9)
- **BellLog** (c.f., Equation 3.10)
- **CF-IDF** (c.f., Equation 3.5)
- **HCF-IDF** (c.f., Equation 3.11)
- **BM25C** (c.f., Equation 3.6)
- **BM25HC** (c.f., Equation 3.12)

### 4.2.2 Usage of Older Scientific Publications

Scientific publications are released in different years. To investigate the influence of older scientific publications on researcher profiles, in our experiment we create researcher profiles with respect to three sets of scientific publications that are split by time. How the scientific publications are split depends on datasets. In addition, we examine how the recommendation performance varies as older scientific publications are incrementally added to researcher profiles.

### 4.2.3 Content Richness

Researchers' profiles are constructed based on their own scientific publications. However, the available contents of scientific publications differ. Therefore, we compare the recommendation performance when profiles are constructed based on only titles, and on both titles and abstracts.

**Title** The researcher profiles are created based on only titles of their own scientific publications.

**All (Title + Abstract)** The researcher profiles are constructed based on both titles and abstracts of their own scientific publications.

In addition, users and researchers have different numbers of social media items and their own scientific publications, respectively. In the experiment, we also analyze how these different numbers influence recommendation performance.

### 4.2.4 Similarity Function

We calculate the similarity scores between a user profile $\Phi(J_u)$ and each of the researcher profiles $\Phi(d)$. All profiles are represented as a term weight vector or an entity weight vector. As similarity function $\sigma$, we use the cosine similarity, which has been widely used.

$$\sigma_{cos}(\Phi(J_u), \Phi(d)) = \frac{\Phi(J_u) \cdot \Phi(d)}{\|\Phi(J_u)\| \cdot \|\Phi(d)\|}. \tag{4.1}$$

## 4.3 Experiment

In the experiment, a recommender system suggests researchers based on a user's social media items. Researchers (i .e., candidate items for the recommender system) are profiled based on their own scientific publications. We choose Twitter as a social media platform due to its popularity in the scientific community [LPB+10]. Throughout the experiment, we investigate (i) effectiveness of different profiling methods; (ii) influence of older scientific publications (i .e., do older scientific publications enhance recommendation performance?); (iii) influence of the number of social media items (i .e., tweets) and scientific publications; and (iv) influence of abstracts (i .e., do abstracts of scientific publications improve the recommendation performance?).

We first describe the procedure of the experiment in Section 4.3.1. Subsequently, Section 4.3.2 details the two datasets used in the experiment. Section 4.3.3 introduces a metric used for evaluation.

### 4.3.1 Procedure

We first identify users who have both a Twitter account and a record of scientific publications. Subsequently, we compute social media profiles (i .e., user profiles) and publication profiles (i .e., researcher profiles) for all users. Thereafter, we calculate similarity scores between a user profile and each of the researcher profiles, as described in Section 4.2.4. Researchers are ranked by similarity scores. Please note that a set of researchers also contains the user himself. As ground truth, we consider the user himself (i .e., his own researcher profile) as interesting (i .e., right recommendation). In a practical recommender system, it is not usual for users to find himself as a recommendation. However, we take this approach due to a lack of ground truth and the difficulty of obtaining it [ZL15]. We assume that researchers ranked near the user have similar interests.

### 4.3.2 Datasets

In the experiment, two datasets are used, which are from computer science and medicine, respectively. Twitter is chosen as a social media platform because of its predominance among different social media platforms and its strong use among researchers to disseminate their scientific thoughts [LPB+10]. We introduce the two datasets below.

**Computer Science** We use 88 Twitter accounts in the field of computer science. To identify these accounts, we follow the data collection methodology of Grosse-Bölting et al. [GNS15a]. Specifically, we first retrieve tweets that mention one of the 26 A*-rated[2] computer science conference hashtags via

[2] CORE ranking from 2014, see `http://103.1.187.206/core/`, last accessed on 08/30/2017

Twitter API. A*-rated conferences are chosen because of their importance. We use only the hashtags that are officially employed on the conference web pages or official conference Twitter accounts. Subsequently, we filter the obtained Twitter accounts and keep only Twitter accounts that also have a publication record in DBLP[3]. Although conference hashtags are not necessarily unique, we assume that accounts that have publication records of DBLP use the hashtags to indicate computer science conferences. Through this procedure, we identify 88 Twitter accounts with corresponding DBLP records. Then, we retrieve their tweets using Twitter API. We can obtain $3,200$ tweets at most for each account due to the limitations of the Twitter API. Please note that we use only tweets in English. A user has published 697.58 tweets on average (SD: 443.17) in English.

To obtain each user's scientific publications, we use the extended DBLP dataset[4]. From the dataset, we obtain titles and abstracts of scientific publications authored by one of the 88 users. In total, we obtain $1,059$ publications, where 325 have abstracts. On average a user has 12.03 publications (SD: 13.45). On average 3.69 (SD: 5.12) of them have abstracts. However, the scientific publications of 29 of the 88 users have no abstract. The average published year is 2006.74 (SD: 4.94). The latest publication dates from 2013, and the oldest one from 1983.

As a domain-specific hierarchical knowledge graph, the ACM Computer Classification System (ACM CCS)[5] is employed. The ACM CCS contains $2,299$ entities in the field of computer science as well as their relations, and $9,086$ labels. On average, an entity has 4.95 labels (SD: 3.59). According to Kapanipathi et al. [KJV+14], the number of entities over the different levels in a hierarchical knowledge graph follows a normal distribution for applying Bell and BellLog. We verified this by visual inspection of the ACM CCS.

**Medicine** In addition to the domain of computer science, we also conduct the experiment in the domain of medicine. We use 64 Twitter accounts who have a publication record. These Twitter users are identified by searching the top five journals[6] on Twitter. Specifically, we query each of the five journal hashtags using the Twitter API and extract Twitter users who use at least one of those hashtags. Subsequently, we filter the obtained users

[3] `http://dblp.uni-trier.de/`, last accessed on 08/30/2017

[4] AMiner Citation Network Dataset, `http://arnetminer.org/lab-datasets/citation/DBLP_citation_Sep_2013.rar`, last accessed on 08/30/2017

[5] `http://www.acm.org/about/class/class/2012`, last accessed on 08/30/2017

[6] `http://impactfactor.weebly.com/medicine.html`, last accessed on 06/02/2015

and keep only the users who also have a record in the PubMed database[7], resulting in our 64 Twitter accounts. On average, a user has published 1508.13 tweets (SD: 1282.62) in English.

To obtain the publications of the 64 users, we access the PubMed database. We obtain publications via API called E-utility functions[8]. On average a user has 50.34 publications (SD: 65.95). On average 43.27 (SD: 60.23) of them have abstracts. However, 4 of the 64 users have no abstract. The average year of publication is 2010.40 (SD: 3.64). The latest publication dates from 2015, and the oldest from 1976.

As a hierarchical domain-specific knowledge graph, we use the Medical Subject Headings (MeSH)[9][10]. The MeSH contains $27,300$ entities in the field of medicine as well as their relations, and $224,368$ labels. Thus, on average, an entity has 8.22 labels (SD: 9.19). A visual inspection confirmed that the number of entities over the different levels follows a normal distribution.

### 4.3.3 Metric

As metric, we use the Mean Reciprocal Rank (MRR) as follows:

$$MRR = \frac{1}{|U|} \sum_{u \in U} \frac{1}{rank(d_u)}, \tag{4.2}$$

where $rank(d_u)$ denotes the rank at which researcher profile $d_u$ corresponding a user $u$ appears in the list of all researcher profiles sorted by similarity scores.

## 4.4 Results

In this section, we report the results of the experiment. We start with the influence of the different profiling methods. Subsequently, Section 4.4.2 reports the influence of the older scientific publications. Finally, Section 4.4.3 shows the impact of the numbers of social media items and scientific publications, as well as the influence of using abstracts for profiling researchers.

[7] https://www.ncbi.nlm.nih.gov/pubmed, last accessed on 08/30/2017

[8] http://www.ncbi.nlm.nih.gov/books/NBK25500/, last accessed on 08/30/2017

[9] 2015 MeSH “Descriptor Records” retrieved 05/16/2015, http://www.nlm.nih.gov/mesh/filelist.html

[10] We convert the original .xml file into the .nt file using the convertor HIVE https://code.google.com/p/hive-mrc/, last accessed on 08/30/2017

Table 4.1: MRR of profiling methods (SD in parentheses). The best overall results are marked in bold font.

| | **Computer Science** | | **Medicine** | |
|---|---|---|---|---|
| | Title | All | Title | All |
| TF-IDF | .31 (.35) | **.33** (.37) | .38 (.42) | .38 (.42) |
| BM25 | **.33** (.38) | .32 (.40) | .33 (.39) | .33 (.39) |
| Freq | .18 (.28) | .21 (.29) | .25 (.36) | .26 (.36) |
| Basic | .15 (.28) | .17 (.30) | .25 (.36) | .28 (.38) |
| Bell | .18 (.30) | .21 (.32) | .23 (.34) | .25 (.36) |
| BellLog | .18 (.28) | .21 (.29) | .25 (.36) | .26 (.36) |
| CF-IDF | .22 (.30) | .24 (.32) | .38 (.41) | .38 (.41) |
| HCF-IDF | .22 (.31) | .22 (.31) | .37 (.41) | .38 (.42) |
| BM25C | .26 (.35) | .25 (.34) | **.43** (.44) | .38 (.42) |
| BM25HC | .24 (.33) | .25 (.35) | .41 (.42) | **.40** (.43) |

### 4.4.1 Influence of Profiling Methods

Table 4.1 illustrates the recommendation performance of each profiling method. While BM25 and TF-IDF perform best in the computer science dataset, BM25C and BM25HC outperform the other profiling methods in the medicine dataset. In terms of the difference between the two academic domains, the medicine dataset consistently shows better recommendation performance. The medicine dataset contains fewer users (i .e., 64 users) than the computer science dataset (i .e., 88 users). Thus, the minimum value of reciprocal rank (i .e., 1/64) in the medicine dataset is higher than the one in the computer science dataset (i .e., 1/88). Therefore, the medicine dataset naturally achieves higher MRR than the computer science dataset does.

We investigate if there are significant differences between the profiling methods. No significance is revealed in the medicine dataset. On the other hand, for the computer science dataset, BM25 and TF-IDF are significantly different compared to Freq, Basic, Bell, and BellLog ("titles": $t(87)$ is in $[3.92, 3.34]$, $p < .05$, $\epsilon = .36$, "all": $t(87)$ is in $[4.21, 3.32]$, $p < .05$, $\epsilon = .35$). Thus, there is no significant difference between term-based profiling methods and profiling methods using knowledge graphs that involve statistical methods.

### 4.4.2 Influence of Usage of Older Scientific Publications

Scientific publications in both datasets originate from various years. We assume that researcher profiles based on newer scientific publications are more similar to

Table 4.2: MRR of the three periods of scientific publications for the computer science dataset.

| | **Computer Science** | | |
|---|---|---|---|
| | **1983-2005** | **2006-2009** | **2010-2013** |
| TF-IDF | .47 (.39) | .57 (.39) | .76 (.35) |
| BM25 | .43 (.38) | .49 (.42) | .71 (.41) |
| Freq | .33 (.32) | .46 (.40) | .43 (.38) |
| Basic | .34 (.35) | .40 (.39) | .32 (.29) |
| Bell | .34 (.32) | .47 (.42) | .37 (.37) |
| BellLog | .33 (.32) | .46 (.40) | .42 (.38) |
| CF-IDF | .36 (.34) | .43 (.31) | .38 (.33) |
| HCF-IDF | .37 (.33) | .48 (.33) | .38 (.33) |
| BM25C | .48 (.39) | .36 (.30) | .39 (.36) |
| BM25HC | .40 (.36) | .41 (.32) | .44 (.39) |

corresponding user profiles based on social media items. To verify this assumption, we compare the recommendation performance using three scientific publication sets in each of the two datasets. Tables 4.2 and 4.3 show the resulting MRR with respect to each period for the computer science dataset and the medicine dataset, respectively. We observe that profiles based on newer scientific publications demonstrate better MRR.

In addition, we examine how the older scientific publications influence recommendation performance. Specifically, we start to measure MRR with researcher profiles based on scientific publications in the most recent year and incrementally add scientific publications that are published in the older years. We observe how MRR changes as we add older scientific publications for researcher profiles. Figures 4.1 and 4.2 illustrate the results of the experiments for the computer science dataset and the medicine dataset, respectively. For the profiling methods using knowledge graphs, we observe that the recommendation performs best when using all scientific publications dating from after around 2004 in the computer science dataset. However, BM25HC demonstrates the best recommendation performance with all scientific publications published after 2000. For TF-IDF and BM25, the recommendation performs best when considering all scientific publications published after around 2010. When using scientific publications published before 2010, the recommendation performance gets worse, especially for BM25. In contrast, for the medicine dataset, the recommendation performance does not vary much when older scientific publications are added to compute researcher profiles. However,

Table 4.3: MRR of the three periods of scientific publications for the medicine dataset.

| | **Medicine** | | |
|---|---|---|---|
| | **1976-2008** | **2009-2012** | **2013-2015** |
| TF-IDF | .44 (.41) | .52 (.46) | .56 (.45) |
| BM25 | .47 (.44) | .49 (.43) | .52 (.45) |
| Freq | .30 (.39) | .36 (.40) | .41 (.44) |
| Basic | .31 (.39) | .39 (.43) | .35 (.42) |
| Bell | .28 (.36) | .35 (.39) | .40 (.43) |
| BellLog | .30 (.39) | .36 (.40) | .41 (.44) |
| CF-IDF | .41 (.43) | .48 (.45) | .51 (.45) |
| HCF-IDF | .41 (.43) | .47 (.44) | .50 (.45) |
| BM25C | .39 (.41) | .45 (.42) | .54 (.45) |
| BM25HC | .42 (.41) | .46 (.42) | .54 (.45) |

for TF-IDF and BM25, the recommendation performance is low when using only scientific publications from the most recent year.

### 4.4.3 Influence of Content Richness

As shown in Table 4.1, we observe that abstracts have a positive influence on the profiling methods Freq, Basic, Bell, and BellLog. On the other hand, for the profiling methods TF-IDF, BM25, CF-IDF, HCF-IDF, BM25C, and BM25HC that involve statistical methods, abstracts have almost no influence, or a negative influence.

Furthermore, we investigate whether the number of tweets (i .e., social media items) and scientific publications have an influence on the recommendation performance. We compute the correlations between MRR and the number of tweets and scientific publications using the Kendall rank coefficient. Tables 4.4 and 4.5 present the results. While we observe a moderate correlation between MRR and the number of scientific publications as shown in Table 4.5, Table 4.4 indicates that there is almost no correlation between MRR and the number of tweets.

## 4.5 Discussion

In terms of the profiling methods, the statistical methods TF-IDF, BM25, BM25C, and BM25HC demonstrate overall better recommendation performance, as shown in Table 4.1. Table 4.1 demonstrates that while TF-IDF and BM25 perform best in the computer science dataset, BM25C and BM25HC outperform the others

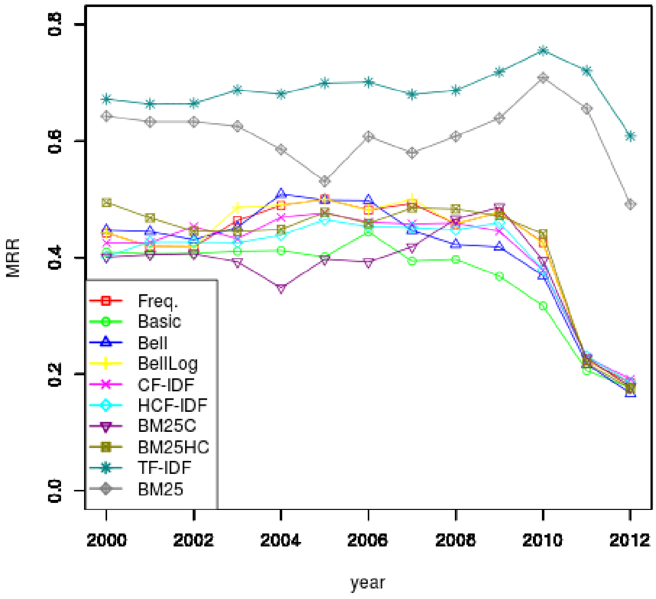


Figure 4.1: Influence of the older scientific publications on recommending researchers in the computer science dataset. All scientific publications published after a year shown in the x-axis are used for researcher profiles. The y-axis represents the MRR.

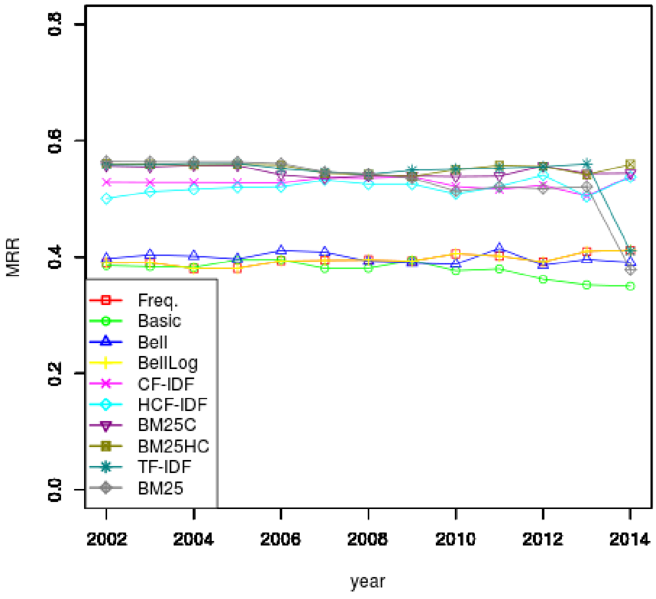


Figure 4.2: Influence of the older scientific publications on recommending researchers in the medicine dataset. All scientific publications published after a year shown in the x-axis are used for researcher profiles. The y-axis represents the MRR.

Table 4.4: Kendall rank coefficient between MRR and the number of tweets. The $p$-values in parentheses are marked in bold font if $\leq .05$.

| | **Computer Science** | | **Medicine** | |
|---|---|---|---|---|
| | title | all | title | all |
| TF-IDF | -.01 (.91) | -.03 (.65) | .01 (.94) | .03 (.72) |
| BM25 | .01 (.85) | .01 (.92) | -.01 (.95) | .00 (.97) |
| Freq | .07 (.36) | .00 (.99) | -.01 (.89) | .00 (.98) |
| Basic | .11 (.13) | .03 (.70) | -.05 (.59) | .00 (.96) |
| Bell | .10 (.20) | .00 (.96) | -.05 (.62) | -.02 (.82) |
| BellLog | .10 (.20) | .00 (.99) | -.02 (.79) | -.01 (.92) |
| CF-IDF | .09 (.42) | .02 (.91) | .07 (.97) | .05 (.87) |
| HCF-IDF | .10 (.35) | .03 (.82) | .04 (.87) | .02 (.69) |
| BM25C | .12 (.28) | .06 (.84) | .06 (.98) | .03 (.49) |
| BM25HC | .16 (.22) | .08 (.91) | .00 (.62) | .01 (.35) |

Table 4.5: Kendall rank coefficient between MRR and the number of scientific publications. The $p$-values in parentheses are marked in bold font if $\leq .05$.

| | **Computer Science** | | **Medicine** | |
|---|---|---|---|---|
| | title | all | title | all |
| TF-IDF | .33 (**.00**) | .37 (**.00**) | .37 (**.00**) | .45 (**.00**) |
| BM25 | .36 (**.00**) | .44 (**.00**) | .47 (**.00**) | .53 (**.00**) |
| Freq | .31 (**.00**) | .36 (**.00**) | .49 (**.00**) | .48 (**.00**) |
| Basic | .27 (**.00**) | .37 (**.00**) | .53 (**.00**) | .54 (**.00**) |
| Bell | .29 (**.00**) | .37 (**.00**) | .51 (**.00**) | .50 (**.00**) |
| BellLog | .33 (**.00**) | .38 (**.00**) | .49 (**.00**) | .48 (**.00**) |
| CF-IDF | .24 (**.00**) | .31 (**.00**) | .41 (**.00**) | .44 (**.00**) |
| HCF-IDF | .28 (**.00**) | .36 (**.00**) | .43 (**.00**) | .45 (**.00**) |
| BM25C | .20 (**.00**) | .24 (**.00**) | .43 (**.00**) | .48 (**.00**) |
| BM25HC | .23 (**.00**) | .28 (**.00**) | .45 (**.00**) | .50 (**.00**) |

in the medicine dataset. A possible reason is the richness of the domain-specific knowledge graph. While the ACM CCS for the computer science dataset contains only $2,299$ entities with $9,068$ labels, the MeSH for the medicine dataset has $27,300$ entities with $224,386$ labels. Thus, the MeSH covers many more entities and labels, which enable to extract sufficient entities to generate user profiles.

In terms of the use of older scientific publications, Figure 4.1 shows a negative influence on the recommendation performance in the computer science dataset. Thus, we should take into account temporal aspects for researcher profiles. On the other hand, Figure 4.2 indicates less influence of the older publications in the medicine dataset. A possible reason is that researchers working in the field of medicine might be less likely to change their professional interests than researchers in the field of computer science. In addition, the terminology in medicine is much more stable and less agile than in computer science, where new "buzzwords" emerge every six months.

Regarding the influence of the number of scientific publications, we observe a moderate correlation. The correlations between the recommendation performance and the number of tweets are weaker than those between the recommendation performance and the number of scientific publications. A possible reason is that users disseminate not only tweets that are relevant to their professional interests, but also those that are unrelated to them, e .g., private travels. In contrast, titles and abstracts of scientific publications contain only professional interests. Thus, we observe a weaker correlation between recommendation performance and the number of tweets compared to the number of scientific publications. Finally, the result shows that abstracts slightly improve the recommendation performance.

# Chapter 5

# Application II: Recommender System of Scientific Publications

The second application is the recommender system for scientific publications. It suggests interesting publications to users based on a user's social media items. Recommending based on social media items has two advantages. First, users receive recommendations based on their current and ongoing professional interests. Second, it mitigates the cold-start problem observed in collaborative filtering [JZF+10]. The cold-start problem refers to the initial situation in which a recommender system does not yet know anything about a user. In addition, it can work for young researchers such as doctoral students who do not yet have a publication record.

In this chapter, we conduct an online experiment to evaluate the influence of three factors on a recommender system based on a user's social media items. The first factor is *Profiling Method*. For this factor, we compare CF-IDF, HCF-IDF, which are defined in Section 3.2, and LDA [BNJ03]. The second factor is *Temporal Decay Function*. We compare the sliding window function and the exponential function, which are introduced in Section 3.3. Finally, the third factor is *Publication Content*, for which we investigate the influence of the richness of content used for profiling candidate items (i .e., scientific publications). We compare the profiles based on both full texts and titles with those based on titles only. In total, we compare twelve recommendation strategies by making use of different combinations of the three experimental factors. We choose Twitter as social media platform due to its popularity in the scientific community [LPB+10]. We use the corpus of the scientific publications in the field of economics as candidate items. We have recruited 123 subjects who have worked in economics and have posted about their professional interests on Twitter.

The experiment demonstrates that the recommendation strategy that employs CF-IDF and the sliding window function based on both titles and full texts achieves the overall best recommendation performance. Although it shows the highest performance, however, it has the drawback that it requires the full texts of scientific publications. It is remarkable that the recommendation strategies with HCF-IDF can achieve comparable recommendation performance using only titles. In fact, a statistical analysis finds no significant difference between the best performing strategy and recommendation strategies with HCF-IDF. Therefore, we conclude that HCF-IDF can mitigate the sparseness and shortness of titles. This is a promising insight since full texts of scientific publications are frequently unavailable, e .g., due to legal reasons.

We first formalizes the problem in Section 5.1. Subsequently, Section 5.2 describes the three experimental factors. Section 5.3 explains the experiment procedure and setup. Section 5.4 reports the results. Finally, Section 5.5 discusses the results.

## 5.1 Problem Statement

We tackle the problem of recommending scientific publications based on a user's social media items. The problem can be decomposed into three parts:

**Profiling a user** A user $u$ generates social media items $J_u$. A user profile is provided by $\Phi(J_u)$ as Definition 3.2.

**Profiling scientific publications** We have scientific publications $d \in D$ as candidate items. A publication profile is provided by $\Phi(d)$ as Definition 3.1.

**Ranking scientific publications for a user** Scientific publications are ranked based on similarity scores between the user profile $\Phi(J_u)$ and the publication profile $\Phi(d)$. A function that outputs these similarity scores is defined as $\sigma : \Phi(J_u), \Phi(d) \rightarrow [0, 1]$. The recommender system computes similarity scores between a user profile $\Phi(J_u)$ and each of the publication profiles $\Phi(d)$. Publications whose similarity scores are ranked in the top-$k$ are recommended.

## 5.2 Design of the Experimental Factors

We investigate three experimental factors in the experiment: *Profiling Method*, *Temporal Decay Function*, and *Publication Content*. Table 5.1 illustrates the

Table 5.1: The three experimental factors and their choices for the experiment span a total of $3 \times 2 \times 2 = 12$ recommendation strategies.

| Factor | Possible Design Choices | | |
|---|---|---|---|
| *Profiling Method* | CF-IDF | HCF-IDF | LDA |
| *Temporal Decay Function* | Sliding Window | Exponential | |
| *Publication Content* | Title | All (title + full-text) | |

design space of the experiment, where each cell is a possible design choice we can make. We detail the factor *Profiling Method* in Section 5.2.1, *Temporal Decay Function* in Section 5.2.2, and *Publication Content* in Section 5.2.3. Subsequently, we describe similarity functions $\sigma$ in Section 5.2.4.

### 5.2.1 Profiling Methods

We investigate three profiling methods to construct user profiles and publication profiles. The experiment focuses on the profiling methods using knowledge graphs, which are introduced in Section 3.2.2. In particular, we assume the use of a domain-specific hierarchical knowledge graph.

**CF-IDF** CF-IDF as defined in Equation 3.5 counts frequencies of an entity instead of frequencies of a term. For computing CF-IDF for social media items $j \in J_u$, we replace $d$ and $D$ in Equation 3.5 with $j$ and $J_{rdm}$, respectively. $J_{rdm}$ is a set of random social media items and allows us to better distinguish relevant entities in the set of the user's social media items $J_u$, as Chen et al. [CNN+10] and Lu et al. [LLZ12] did with TF-IDF. For instance, assuming that there are two social media items from a user $u$ and both include an entity referring to "currency competition," this entity should have a high weight in the user profile. However, IDF and CF-IDF would be 0 because the entity is so common in the user's social media items.

The random social media items are sampled from public postings. In the experiment, we obtain them from the public Twitter stream using the Twitter API. We first conduct a simple pre-experiment to empirically determine the optimal amount of random social media items to use for user profiles in the context of recommending scientific publications. Given the results of this pre-experiment, we set the size of random social media items to $|J_{rdm}| = 5 \cdot |J_u|$. In the pre-experiment, we apply different sizes of random social media items $J_{rdm}$, starting from 0 to 1000. For 26 Twitter accounts, we compute the IDF for user profiles over $J_u \cup J_{rdm}$ with different sizes of random social media items $J_{rdm}$. Then, we compare it using cosine similarity with the user profile computed only over $J_u$. The 26 Twitter accounts are taken from a list of famous economists[1] who frequently tweet. We ensure that the set of random social media items $J_{rdm}$ is disjoint from the user's social media items, i. e.. $J_{rdm} \cap J_u = \emptyset$. In this pre-experiment, we examine the changes of the cosine similarity while adding more random social media items. We observe that the changes in the IDF become stable after about $5 \cdot |J_u|$. The changes indicate the influence of the IDF on user profiles. Please note that this result may depend on the domain of economics. Thus, a different size of random social media items may be chosen for other domains.

For publication profiles, CF-IDF is computed as Equation 3.5.

**HCF-IDF** The advantage of HCF-IDF, which is defined in Equation 3.11, is that it combines the statistical strength of CF-IDF with semantics provided by the hierarchical knowledge graph. HCF-IDF for a social media item $j$ is computed by replacing $d$ and $D$ in Equation 3.11 with $j$ and $J_{rdm}$, respectively. Similar to CF-IDF, random social media items $J_{rdm}$ are employed.

For publication profiles, HCF-IDF is computed as Equation 3.11.

**LDA** As the third profiling method, we use LDA [BNJ03], an unsupervised topic modeling method. LDA identifies latent topics in a document corpus, where each document is represented as a probability distribution over topics, and in turn each topic is represented as a probability distribution over terms. We treat the set of social media items $J_u$ published by a user $u$ as one *single* social media document in this profiling method, because it is known that topic models that treat a user's microblog postings as one combined social media document outperform topic models computed over single postings for recommendation tasks [HD10]. We first create a topic model for the entire publication corpus $D$. Subsequently, we run LDA with the given topic model of the publication corpus $D$ to infer a probability distribution over topics for a user's social media document $J_u$. The details of the hyper parameters and tools are described in Section 5.3.3. We treat each topic generated by LDA as an entity $r$. The relevance of a topic $r$ in user profiles is defined by:

$$\nu_{lda}(r, J_u) = p(r \mid J_u). \tag{5.1}$$

[1] `http://www.huffingtonpost.com/2012/11/13/economists-twitter_n_2122781.html`, last accessed on 08/31/2017

The relevance in publication profiles is computed as:

$$\nu_{lda}(r, d) = p(r \mid d). \tag{5.2}$$

In Equations 5.1 and 5.2, $p(r \mid J_u)$ and $p(r \mid d)$ denote the probability of the entity (i .e., topic) $r$ in the social media items $J_u$ and scientific publication $d$, respectively.

### 5.2.2 Temporal Decay Functions

In our experiment, we compare two temporal decay functions: the sliding window function and the exponential function, which are introduced in Section 3.3. In the past, both temporal decay functions have been used in recommender systems [SWL+13; OBP12; SK10]. However, so far they have not been empirically compared. The final weights of entities $w(r, j)$ and $w(r, d)$ are computed by combining a functions $\nu$ in the previous section with a temporal decay function $\mu$, as defined in Definition 3.3.

Please note that when employing LDA, the two temporal decay functions can be applied only on scientific publications, because a set of social media items is treated as one single social media document. For social media items, we apply the no temporal decay function which is defined as Equation 3.13.

We describe the two temporal decay functions below.

**Sliding Window** The sliding window function is defined as Equation 3.14. For user profiles, we set the window size based on the work of Orlandi et al. [OBP12], who found that the half-life time of a social media item is 250 days. Hence, we set $t_{window_{social}} = 250$ days. For publication profiles, the sliding window function filters out scientific publications older than $t_{window}$. We choose the window size according to the work of Sangam and Mogali [SM13]. They observed a half-life time of 9.04 years for scientific publications in social science. In the experiment, we use the publication corpus in economics that has a large overlap with social science. Therefore, we set $t_{window_{pub}} = 9.04$ years and filter out scientific publications published more than 9.04 years ago.

**Exponential** The exponential function is defined as Equation 3.15. For user profiles, we set $\tau = 360$ days, since Orlandi et al. [OBP12] observed that the recommendation performance based on user profiles with $\tau = 360$ days was better than the one with $\tau = 120$ days. For publication profiles, we set

$\tau = 13.05$ years because Sangam and Mogali [SM13] found that the mean life of scientific publications in social science is 13.05 years.

### 5.2.3 Publication Content

This factor is used to examine whether it is possible to make reasonable recommendations based on only titles of scientific publications. To this end, we compare two sources for publication profiles.

**Title** The publication profiles are made based on only titles of scientific publications.

**All (Title + Full-text)** The publication profiles are constructed based on both titles and full texts of scientific publications.

### 5.2.4 Similarity Functions

We calculate the similarity score between a user profile $\Phi(J_u, G)$ and each publication profile $\Phi(d, G)$. These profiles are represented as vectors, where each element corresponds to an entity weight.

**Temporal Cosine Similarity** The profiling methods CF-IDF and HCF-IDF employ the temporal cosine similarity as:

$$\sigma_{tcossim}(\Phi(J_u), \Phi(d)) = \mu(t(d)) \cdot \frac{\Phi(J_u) \cdot \Phi(d)}{\|\Phi(J_u)\| \cdot \|\Phi(d)\|}. \tag{5.3}$$

This extends the cosine similarity by a temporal decay function $\mu(t(d))$, which results in higher similarity scores for newer scientific publications. $t(d)$ returns the publication year of a scientific publication $d$.

Regarding HCF-IDF, we also consider the hierarchical cosine similarity by Ganesan et al. [GGW03], which takes into account the hierarchical structure of a knowledge graph when computing similarity scores. However, the pre-experiment reveals that it does not work well for HCF-IDF in terms of the recommendation performance. One of the possible reasons is that broader entities are boosted too much through both the profiling method and the calculation of the similarity score. Therefore, we use the temporal cosine similarity for HCF-IDF.

**Dot Product** LDA employs the dot product, which is defined as:

$$\sigma_{dp}(\Phi(J_u), \Phi(d)) = \Phi(J_u) \cdot \Phi(d). \tag{5.4}$$

Since LDA represents documents as a probability distribution over topics, the Kullback-Leibler divergence (KL divergence) is considered as a more reasonable similarity function in general. However, Hazen [Haz10] reported that the dot product outperformed both the cosine similarity and the KL divergence when representing documents as a probability distribution over topics using LDA. For this reason, the dot product is chosen.

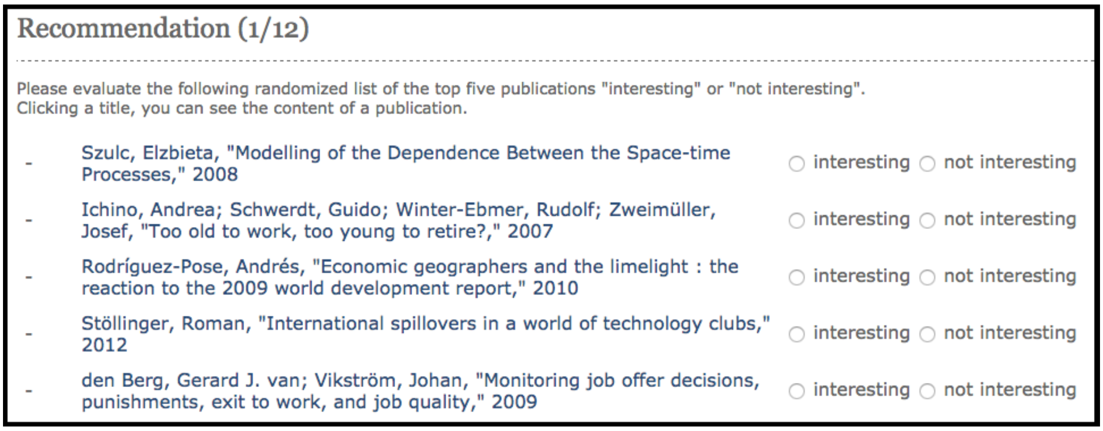
## 5.3 Experiment

We conduct an online experiment with 123 subjects to identify the best recommendation strategy regarding the factors described in Section 5.2. We choose Twitter as social media platform since it has been widely used in scientific communities [LPB+10]. We design the experimental setup and procedure following the work of Chen et al. [CNN+10], where each subject receives top-5 recommendations for each of the twelve recommendation strategies formed from the three experimental factors. The recommendation performance of each strategy is measured using rankscore [BHK98] following Bostandjiev et al. [BOH12]. We describe the details of the experiment procedure in Section 5.3.1 and the subjects in Section 5.3.2. Subsequently, Section 5.3.3 describes the dataset and the knowledge graph used in the experiment. Finally, Section 5.3.4 introduces the evaluation metric.

### 5.3.1 Procedure

Subjects are invited to a web application where the twelve recommendation strategies are implemented. On this application, the subjects first input their public Twitter handle and e-mail address. Then, their tweets are retrieved by the Twitter API. The extracted tweets are used to construct user profiles with each of the three profiling methods and two temporal decay functions. Based on the user profiles, personalized top-$k$ recommendations of scientific publications are generated using each strategy. We set the number of recommendations per strategy to $k = 5$, following Chen et al. [CNN+10]. After computing the recommendations, the subjects receive an e-mail invitation to assess the recommendations. Since we employ a repeated-measure design, the subjects go through all twelve recommendation strategies as it was conducted by Chen et al. [CNN+10]. Thus, each subject obtains $12 \cdot 5 = 60$ recommendations in total throughout the experiment.

Prior to starting the experiment, subjects are informed about the task of the experiment, i .e., rating the recommended publications based on their research interests, and confirm their consent. On each of the subsequent pages, the subjects



Recommendation (1/12)

Please evaluate the following randomized list of the top five publications "interesting" or "not interesting".
Clicking a title, you can see the content of a publication.

- Szulc, Elzbieta, "Modelling of the Dependence Between the Space-time Processes," 2008 ○ interesting ○ not interesting
- Ichino, Andrea; Schwerdt, Guido; Winter-Ebmer, Rudolf; Zweimüller, Josef, "Too old to work, too young to retire?," 2007 ○ interesting ○ not interesting
- Rodríguez-Pose, Andrés, "Economic geographers and the limelight : the reaction to the 2009 world development report," 2010 ○ interesting ○ not interesting
- Stöllinger, Roman, "International spillovers in a world of technology clubs," 2012 ○ interesting ○ not interesting
- den Berg, Gerard J. van; Vikström, Johan, "Monitoring job offer decisions, punishments, exit to work, and job quality," 2009 ○ interesting ○ not interesting

Figure 5.1: Screenshot of the evaluation page. It shows a list of top-5 recommendations in randomized order for the first of twelve recommendation strategies, which are also randomly ordered. For each recommendation, the subjects see its bibliographic record. In addition, they can see the original PDF files by clicking on the bibliographic record. The subjects rate each recommended publication as "interesting" or "not interesting" based on their professional interests.

see a list of five recommendations produced by one of the twelve recommendation strategies. An example screenshot of the evaluation page is shown in Figure 5.1. For each recommended publication, the subjects see its bibliographic information, i .e., authors, title, and year of publication. In addition, the subjects can look into the original PDF files by clicking on a link attached to the bibliographic record.

To avoid bias, the subjects go through the twelve recommendation strategies in random order. In addition, the five recommendations on each recommendation list are shown in random order to avoid the well-known ranking bias, i .e., subjects typically assume that top-ranked items are essentially more relevant [BOH12; CNN+10]. However, the true ranks of the recommendations as well as the positions where they appeared on the evaluation page are stored in the database for later analysis. Prior to starting the experiment, we explicitly inform the subjects that we have randomized the order of the twelve recommendation strategies and the scientific publications in the recommendation lists.

The subjects evaluate each recommendation as "interesting" or "not interesting" by clicking on radio buttons next to the publication records, as in Chen et al.'s [CNN+10] experiment. Please note that the subjects have to evaluate all recommended publications. Thus, they cannot skip the evaluation for any recommended publications.

At the end of the experiment, we collect the subjects' demographic information including gender, age, highest academic degree, major, years of profession, and current employment status (academia or industry). Finally, subjects can make free comments regarding the experiment.

### 5.3.2 Subjects

We recruit 123 subjects through mailing lists, tweets, and word-of-mouth on the Internet. Initially, 160 subjects registered their Twitter handle and e-mail address for the experiment. Among them, 134 subjects started the experiment after receiving the e-mail invitation. Of these 134 subjects, only eleven dropped out in the course of assessing the recommendations.

Thus, we obtain evaluations for all of the recommendation strategies from 123 subjects. Among them, 27 subjects are female. The average age of the subjects is 32.83 years (SD: 7.34). Regarding the highest academic degree, 21 subjects have a bachelor's degree, 58 a master's, 32 a PhD, and 12 are lecturers or professors. While 83 subjects work in academia, 40 work in industry. The subjects' tweets are retrieved via Twitter API, which allows us to retrieve a maximum of $3,200$ tweets per subject. Only tweets in English are collected, since the scientific publications are also in English. The subjects have published on average $1,096.82$ English tweets (SD: $1,048.46$). The maximum and minimum numbers of tweets are $3,192$ and 2, respectively. Twitter users who have not produced any tweets in the last 250 days cannot register and participate in the experiment, since we use $t_{window_{social}} = 250$ days for the temporal decay function sliding window (see Section 5.2.2). Five Twitter users could not participate in the experiment for this reason.

The subjects spend on average 517.54 seconds (SD: 376.72) to complete the evaluation of the $12 \cdot 5 = 60$ recommendations. This does not include the time spent to register for the experiment, read the instructions, and fill out the final questionnaire.

As an incentive, each subject receives information about his other most similar economists among 26 famous economists[2]. In addition, the subjects are shown the top-5 dominant entities in his tweets after the experiment. Furthermore, the subjects could opt-in to a raffle for one of two Amazon vouchers worth 50 €.

[2] `http://www.huffingtonpost.com/2012/11/13/economists-twitter_n_2122781.html`, last accessed on 08/31/2017

### 5.3.3 Dataset and Preprocessing

As candidate items, we use a large corpus of scientific publications in economics. We employ a high-quality thesaurus as a knowledge graph for profiling methods. In addition, in this section we explain how to process tweets and scientific publications, and we describe an implementation of LDA.

**Corpus of scientific publication** We collaborate with the providers of EconBiz[3], a portal for scientific publications in economics. EconBiz is managed by ZBW, the German National Library of Economics. From this portal, we obtain 1 million URLs of open access publications and extract full texts as well as metadata (i .e., authors, title, year of publication) from $413,098$ scientific publications. Finally, we determine the language used in each publication using a language detection library for Java[4]. The details of the language detection library[5] are documented online. We determine that $279,381$ of $413,098$ scientific publications are in English. Therefore, we use these $279,381$ scientific publications in the experiment.

**Knowledge graph** We use the STW (Standard Thesaurus for Economics)[6] as a domain-specific hierarchical knowledge graph for profiling methods. The STW is a thesaurus specialized for economics and manually maintained by domain experts of the ZBW. Thus, it is of high quality. The knowledge graph is poly-hierarchically organized with six levels. It contains $6,335$ entities and $11,679$ labels. The hierarchically organized entities are connected with each other via $14,875$ relations (i .e., `broader` or `narrower`). To extract as many labels as possible, we enrich the original STW with DBpedia redirects[7]. From DBpedia redirects, we can retrieve the synonymous labels for an entity. STW contains $2,692$ entities that have both a DBpedia mapping and one or more DBpedia redirects. For example, for the entity "Telecommunications industry" in STW, we obtain the DBpedia redirects "Telecommunications operator" and "Telephone companies," and use them as synonymous labels referring to the entity "Telecommunications industry." Finally, the extended STW contains $6,335$ entities and $37,733$ labels. This extended STW is used for the profiling methods CF-IDF and HCF-IDF. For CF-IDF, we ignore the relations between entities.

[3] `http://www.econbiz.de/`, last accessed on 08/31/2017

[4] `https://github.com/shuyo/language-detection`, last accessed on 08/31/2017

[5] `http://www.slideshare.net/shuyo/language-detection-library-for-java`, last accessed on 08/31/2017

[6] `http://zbw.eu/stw/version/8.12/about.en.html`, last accessed on 08/31/2017

[7] `http://oldwiki.dbpedia.org/Downloads39\#redirects`, last accessed on 08/31/2017

**Processing tweets and scientific publications** Here, we describe how we process tweets and scientific publications and how we extract entities from them. We first lemmatize both the tweets and the scientific publications using Stanford Core NLP[8] and remove stop words. Some tweets contain hashtags indicating topics (e.g., #election) and user mentions (e.g., @UNICEF). We remove only the symbols # and @ from these tweets, because Feng and Wang [FW14] observed that the combination of tweets' textual content with the hashtags and user mentions resulted in the highest performance for tag recommendation. Thereafter, we extract entities from the tweets and the scientific publications by matching them with the labels from the extended STW. This process extracts only the users' professional interests and helps to avoid noise (i.e., topics not relevant to professional interests in economics), since we employ a domain-specific knowledge graph. A subject has published on average $1,096.82$ tweets (SD: $1,048.46$). On average, $1,214.93$ entities (SD: $1,181.43$) are contained in a subject's tweets, and 1.07 entities (SD: 0.31) are contained per tweet. We also calculate the ratio of the number of tweets containing at least one entity and the total number of tweets for each subject. This indicates the percentage of tweets that have contributed to creating the user profile. On average, 62.24% of the tweets (SD: 13.55) contain at least one entity from the knowledge graph in economics. These tweets are assumed to be relevant to users' professional interests.

**LDA** To generate the topic model, we first run LDA over the corpus of scientific publications. Following Blei and Lafferty [BL06], we lemmatize the scientific publications using Stanford NLP Core. Subsequently, we remove stop words and terms that appear in fewer than 25 different scientific publications. We optimize the number of topics $K$ regarding the maximum mean log likelihood as suggested by Griffiths and Steyvers [GS04]. We experiment with $K = 20$, 50, 100, 200, 500, 1000, and 5000, and obtain the highest log likelihood when $K = 100$. Therefore, we set $K = 100$ in the experiment. The topic models are computed over 500 iterations. Regarding the hyper parameters for LDA, we set $\alpha = 0.5$ and $\beta = 0.1$, as suggested by Griffiths and Steyvers [GS04]. To infer a topic distribution over a user's tweets, we run LDA again using the topic model of the corpus of the scientific publication. The topic distribution is computed over 200 iterations. Prior to the inference process, we prepare a user's tweets as a single social media document, as described in Section 5.2.1. As an implementation of LDA, we use JGibbLDA[9]. JGibbLDA uses Gibbs sampling based on the work of Griffiths

[8] `http://nlp.stanford.edu/software/corenlp.shtml`, last accessed on 08/31/2017

[9] `http://jgibblda.sourceforge.net/`, last accessed on 08/31/2017

and Steyvers [GS04]. Gibbs sampling is a Markov chain Monte Carlo (MCMC) algorithm for obtaining a sequence of observations which are approximated from a specified multivariate probability distribution. It is used for Bayesian inference instead of a deterministic algorithm such as the expectation maximization (EM) algorithm and makes the learning process of LDA much faster.

### 5.3.4 Metric

To assess the recommendation performance, we compute the rankscore. The rankscore [BHK98] has been used by Bostandjiev et al. [BOH12]. It posits that each successive recommended item in a recommendation list is less likely to be viewed by users with the exponential function, as defined in Equation 5.5.

$$rankscore' = \sum_{d \in hits} \frac{1}{2^{\frac{rank(d)-1}{\theta-1}}}, \tag{5.5}$$

where $\theta$ denotes a viewing halflife parameter controlling the speed of the exponential function. As suggested by Breese et al. [BHK98], we set $\theta = 5$. $hits$ refers to the set of scientific publications evaluated as “interesting” and $rank(d)$ denotes the rank of a recommended item $d$ in a recommendation list. Please note that $rank(d)$ is the actual rank stored in the database. It is different from the position where a publication $d$ appears in the recommendation list (see Section 5.3.1). Then, the normalized rankscore is computed as:

$$rankscore = \frac{rankscore'}{rankscore_{max}}, \tag{5.6}$$

where the maximum rankscore $rankscore_{max}$ is computed as:

$$rankscore_{max} = \sum_{i=1}^{k} \frac{1}{2^{\frac{i-1}{\theta-1}}}. \tag{5.7}$$

Here, $k$ is the number of recommended items. We set $k = 5$.

In addition to the rankscore, precision, Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), and normalized Discounted Cumulative Gain (nDCG) are also computed. Overall, all of these results are similar to the rankscore. Thus, we omit them for the reason of brevity. The interested reader may refer to the details in Appendix A.

## 5.4 Results

This section reports the results of the experiment and conducts the statistical analyses. The anonymized experimental data is available online[10]. We set a significance level of $\alpha = .05$ for all statistical analyses (please do not confuse this with the hyper parameter $\alpha$ for LDA in Section 5.3.3). We first report the best performing strategy among the twelve investigated ones. Subsequently, we analyze the influence of the three experimental factors in Section 5.4.2. Section 5.4.3 then analyzes the influence of the demographic factors, and Section 5.4.4 reports the influence of the amount of content for user profiles. Thereafter, Section 5.4.5 investigates click rates on the PDF files and Section 5.4.6 reports feedback received from subjects in the experiment. Finally, Section 5.4.7 provides the computation time of the recommendation strategies

### 5.4.1 Best Performing Strategy

Table 5.2 summarizes the mean average of the rankscore of the twelve recommendation strategies sorted in descending order. Overall, the best performing strategy is the strategy CF-IDF × Sliding Window × All. We apply a one-way repeated-measure ANOVA to identify whether there are significant differences between the strategies. Before applying the ANOVA, however, we first need to verify whether the variances of the rankscore of the strategies are equal. This is done by using Mauchly's test, which reveals a violation of sphericity in the recommendation strategies ($\chi^2(65) = 435.90$, $p = .00$). This may lead to positively biased F-statistics, and increases the risk of false positives. To reduce this risk, we apply a Greenhouse-Geisser correction of $\epsilon = .61$ and run a one-way repeated-measure ANOVA. It reveals a significant difference in the recommendation strategies ($F(6.60, 805.33) = 21.98$, $p = .00$). To assess the pairwise differences between the strategies, a post-hoc analysis is conducted. We employ Shaffer's modified sequentially rejective Bonferroni procedure (Shaffer's MSRB procedure) [Sha86], which takes into account the number of different experiment conditions, i. e., the number of recommendation strategies. The result of the post-hoc analysis for the rankscore is presented in Table 5.3. The vertical and horizontal dimensions of Table 5.3 show the eleven-by-eleven comparison of the twelve recommendation strategies. There are various significant differences between the strategies ($p < .05$, marked in bold font). For example, one can observe a significant difference between the strategies CF-IDF × Sliding Window × Title

[10] http://dx.doi.org/10.7802/1224, last accessed on 08/31/2017

Table 5.2: Rankscore of the recommendation strategies in decreasing order. M and SD denote mean and standard deviation, respectively.

| | **Recommendation Strategy** | | | **Rankscore** |
|---|---|---|---|---|
| | **Profiling Method** | **Temporal Decay Function** | **Publication Content** | **M (SD)** |
| 1. | CF-IDF | Sliding Window | All | .59 (.33) |
| 2. | HCF-IDF | Sliding Window | All | .56 (.34) |
| 3. | HCF-IDF | Sliding Window | Title | .55 (.33) |
| 4. | HCF-IDF | Exponential | Title | .52 (.30) |
| 5. | CF-IDF | Exponential | All | .51 (.32) |
| 6. | HCF-IDF | Exponential | All | .49 (.30) |
| 7. | CF-IDF | Exponential | Title | .41 (.29) |
| 8. | CF-IDF | Sliding Window | Title | .39 (.27) |
| 9. | LDA | Exponential | Title | .35 (.31) |
| 10. | LDA | Sliding Window | Title | .33 (.31) |
| 11. | LDA | Exponential | All | .32 (.30) |
| 12. | LDA | Sliding Window | All | .27 (.33) |

and HCF-IDF × Sliding Window × All ($t(122) = 4.77$, $p = .00$). However, there is no significant difference between the recommendation strategies CF-IDF × Exponential × Title and LDA × Sliding Window × Title ($t(122) = 2.43$, n.s., $p = .41$).

### 5.4.2 Influence of the Three Experimental Factors

We analyze the results with respect to each experimental factor. We first apply Mendoza's test [Men80] to check for violations of sphericity against the factors. Mendoza's test is an extension of Mauchly's test to adapt it to multi-way repeated-measure ANOVA. It shows violations of sphericity with the global ($\chi^2(65) = 435.90$, $p = .00$) and the factors *Profiling Method* ($\chi^2(2) = 12.21$, $p = .00$), *Profiling Method × Temporal Decay Function* ($\chi^2(2) = 20.02$, $p = .00$), and *Profiling Method × Publication Content* ($\chi^2(2) = 8.61$, $p = .01$). Thereafter, we run a three-way repeated-measure ANOVA with a Greenhouse-Geisser correction of $\epsilon = .60$ for the global and $\epsilon = .91$ for the factor *Profiling Method*, $\epsilon = .87$ for the factor *Profiling Method × Temporal Decay Function*, and $\epsilon = .93$ for the factor *Profiling Method × Publication Content*. Table 5.4 shows the results of the ANOVA with F-ratio, effect size $\eta^2$, and p-value. The effect size is interpreted as small when $\eta^2 > .02$, medium when $\eta^2 > .13$, and large when $\eta^2 > .26$. The analysis reveals significant differences in all three experimental factors and their contributions, except for the factor *Temporal Decay Function*. For all factors with

Table 5.3: Post-hoc analysis with p-values for pairwise comparison over the twelve recommendation strategies using Shaffer's MSRB procedure. The p-values are marked in bold font if $p < .05$, which indicates a significant difference between the two recommendation strategies. Recommendation strategies are sorted by rankscores as shown in Table 5.2.

| | | | | All | Title | Title | All | All | Title | Title | Title | Title | All | All |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Sliding Window | Sliding Window | Exponential | Exponential | Exponential | Exponential | Sliding Window | Exponential | Sliding Window | Exponential | Sliding Window |
| | | | | HCF-IDF | HCF-IDF | HCF-IDF | CF-IDF | HCF-IDF | CF-IDF | CF-IDF | LDA | LDA | LDA | LDA |
| | | | | 2. | 3. | 4. | 5. | 6. | 7. | 8. | 9. | 10. | 11. | 12. |
| 1. | CF-IDF | Sliding Window | All | .99 | .97 | .72 | .22 | .12 | **.00** | **.00** | **.00** | **.00** | **.00** | **.00** |
| 2. | HCF-IDF | Sliding Window | All | | .99 | .99 | .99 | .99 | **.00** | **.00** | **.00** | **.00** | **.00** | **.00** |
| 3. | HCF-IDF | Sliding Window | Title | | | .99 | .99 | .99 | **.00** | **.00** | **.00** | **.00** | **.00** | **.00** |
| 4. | HCF-IDF | Exponential | Title | | | | .99 | .99 | **.01** | **.00** | **.00** | **.00** | **.00** | **.00** |
| 5. | CF-IDF | Exponential | All | | | | | .99 | **.04** | **.00** | **.00** | **.00** | **.00** | **.00** |
| 6. | HCF-IDF | Exponential | All | | | | | | .12 | **.02** | **.00** | **.00** | **.00** | **.00** |
| 7. | CF-IDF | Exponential | Title | | | | | | | .99 | .99 | .41 | .28 | **.01** |
| 8. | CF-IDF | Sliding Window | Title | | | | | | | | .99 | .84 | .61 | **.03** |
| 9. | LDA | Exponential | Title | | | | | | | | | .99 | .99 | .72 |
| 10. | LDA | Sliding Window | Title | | | | | | | | | | .99 | .99 |
| 11. | LDA | Exponential | All | | | | | | | | | | | .88 |

Table 5.4: Three-way repeated-measure ANOVA with Greenhouse-Geisser correction with F-ratio, effect size $\eta^2$, and p-value.

| **Factor** | **F** | $\eta^2$ | **p** |
|---|---|---|---|
| *Profiling Method* | 58.40 | .48 | **.00** |
| *Temporal Decay Function* | 1.17 | .01 | .28 |
| *Publication Content* | 5.18 | .04 | **.02** |
| *Profiling Method* × *Temporal Decay Function* | 4.63 | .04 | **.01** |
| *Profiling Method* × *Publication Content* | 17.09 | .14 | **.00** |
| *Temporal Decay Function* × *Publication Content* | 4.69 | .04 | **.03** |
| *Profiling Method* × *Temporal Decay Function* × *Publication Content* | 3.35 | .03 | **.04** |

Table 5.5: Rankscores, post-hoc analysis for the factor *Profiling Method* using Shaffer's MSRB procedure, and effect size.

**a) Rankscores**

| **Choice** | **M** | **SD** |
|---|---|---|
| HCF-IDF | .53 | .32 |
| CF-IDF | .48 | .31 |
| LDA | .32 | .31 |

**b) Post-hoc analysis p-values**

| | HCF-IDF | LDA |
|---|---|---|
| CF-IDF | **.00** | **.00** |
| HCF-IDF | | **.00** |

**c) Effect size using Cohen's *d***

| | HCF-IDF | LDA |
|---|---|---|
| CF-IDF | .17 | .50 |
| HCF-IDF | | .67 |

significance, we again apply a post-hoc analysis using Shaffer's MSRB procedure with respect to each factor.

**The Factor *Profiling Method*** Tables 5.5(a), (b), and (c) show the rankscores, the post-hoc analysis for the factor *Profiling Method*, and the effect size, respectively. Table 5.5(a) presents the means and standard deviations of the three profiling methods. Table 5.5(b) shows p-values of each pair. Since Table 5.4 shows that this factor has the largest effect size, we further compute the effect size using Cohen's $d$ for each pair, as shown in Table 5.5(c). The post-hoc analysis reveals significant differences between all pairs of HCF-IDF, CF-IDF, and LDA. Although the recommendation strategy CF-IDF × Sliding Window × All performs best as shown in Table 5.2, the best *Profiling Method* is HCF-IDF, as it performs better under all other factors better than CF-IDF and LDA regarding the other factors.

Table 5.6: Rankscores and post-hoc analysis for the factor *Publication Content* using Shaffer's MSRB procedure.

**a) Rankscores**

| **Choice** | **M** | **SD** |
|---|---|---|
| All | .46 | .21 |
| Title | .43 | .20 |

**b) Post-hoc analysis p-values**

| | Title |
|---|---|
| All | **.02** |

Table 5.7: ANOVA for *Profiling Method* × *Temporal Decay Function* interaction.

| **Factor** | **F** | $\eta^2$ | **p** |
|---|---|---|---|
| *Profiling Method* at Sliding Window | 52.71 | .43 | **.00** |
| *Profiling Method* at Exponential | 26.89 | .22 | **.00** |
| *Temporal Decay Function* at CF-IDF | 3.69 | .03 | .06 |
| *Temporal Decay Function* at HCF-IDF | 2.33 | .02 | .12 |
| *Temporal Decay Function* at LDA | 5.26 | .04 | **.02** |

**The Factor *Publication Content*** Table 5.6 shows the post-hoc analysis for the factor *Publication Content*. The result shows that the recommender systems perform better when using both titles and full texts ($F(1,122) = 5.18$, $p = .02$).

**The Factor *Profiling Method* × *Temporal Decay Function*** Table 5.7 shows the results of the ANOVA regarding the factor *Profiling Method* when a choice of the factor *Temporal Decay Function* is fixed and vice versa. Mendoza's test reveals a violation of sphericity in the factor *Profiling Method* when Sliding Window is used ($\chi^2(2) = 9.26$, $p = .01$) and when Exponential is used ($\chi^2(2) = 11.16$, $p = .00$). Therefore, we run a one-way repeated-measure ANOVA with a Greenhouse-Geisser correction of $\epsilon = .93$ for the first row in Table 5.7, and $\epsilon = .92$ for the second row. We also observe significant differences when a choice of the factor *Temporal Decay Function* is fixed and when LDA is employed. The post-hoc analyses are shown in Tables 5.8, 5.9, and 5.10, respectively. In Tables 5.8 and 5.9, a choice of the factor *Temporal Decay Function* is fixed. The results demonstrate that HCF-IDF performs best, followed by CF-IDF and LDA. Thus, the recommendation performance of HCF-IDF is not influenced by the choice of a temporal decay function. Table 5.10 shows the post-hoc analysis of the factor *Temporal Decay Function* when LDA is employed. It indicates that in this case Exponential performs better than Sliding Window.

Table 5.8: Rankscores and post-hoc analysis for the factor *Profiling Method* at Sliding Window using Shaffer's MSRB procedure.

**a) Rankscores**

| **Choice** | **M** | **SD** |
|---|---|---|
| HCF-IDF | .55 | .33 |
| CF-IDF | .49 | .32 |
| LDA | .30 | .32 |

**b) Post-hoc analysis p-values**

| | HCF-IDF | LDA |
|---|---|---|
| CF-IDF | **.01** | **.00** |
| HCF-IDF | ▨ | **.00** |

Table 5.9: Rankscores and post-hoc analysis for the factor *Profiling Method* at Exponential using Shaffer's MSRB procedure.

**a) Rankscores**

| **Choice** | **M** | **SD** |
|---|---|---|
| HCF-IDF | .51 | .30 |
| CF-IDF | .46 | .31 |
| LDA | .34 | .31 |

**b) Post-hoc analysis p-values**

| | HCF-IDF | LDA |
|---|---|---|
| CF-IDF | **.02** | **.00** |
| HCF-IDF | ▨ | **.00** |

Table 5.10: Rankscores and post-hoc analysis for the factor *Temporal Decay Function* at LDA using Shaffer's MSRB procedure.

**a) Rankscores**

| **Choice** | **M** | **SD** |
|---|---|---|
| Exponential | .34 | .31 |
| Sliding Window | .30 | .32 |

**b) Post-hoc analysis p-value**

| | Exponential |
|---|---|
| Sliding Window | **.02** |

Table 5.11: ANOVA for *Profiling Method* × *Publication Content* interaction.

| Factor | F | $\eta^2$ | p |
|---|---|---|---|
| *Profiling Method* at Title | 26.15 | .21 | **.00** |
| *Profiling Method* at All | 55.28 | .45 | **.00** |
| *Publication Content* at CF-IDF | 32.95 | .27 | **.00** |
| *Publication Content* at HCF-IDF | 0.43 | .00 | .51 |
| *Publication Content* at LDA | 2.06 | .02 | .15 |

Table 5.12: Rankscores and post-hoc analysis for the factor *Profiling Method* at Title using Shaffer's MSRB procedure.

**a) Rankscores**

| Choice | M | SD |
|---|---|---|
| HCF-IDF | .54 | .31 |
| CF-IDF | .40 | .28 |
| LDA | .34 | .31 |

**b) Post-hoc analysis p-values**

| | HCF-IDF | LDA |
|---|---|---|
| CF-IDF | **.00** | **.04** |
| HCF-IDF | ▨ | **.00** |

**The Factor *Profiling Method* × *Publication Content*** Table 5.11 shows the results of the ANOVA regarding the factor *Profiling Method* when a choice of the factor *Publication Content* is fixed and vice versa. We observe a significant difference when a choice is fixed and CF-IDF is employed. Mendoza's test indicates a violation of sphericity in the factor *Profiling Method* when All (i .e., titles and full texts) is used ($\chi^2(2) = 25.24$, $p = .00$). Therefore, we run a one-way repeated-measure ANOVA with a Greenhouse-Geisser correction of $\epsilon = .84$ for the second row in Table 5.11. Table 5.12 presents the post-hoc analysis when Title is selected for the factor *Profiling Method*. We see that HCF-IDF outperforms others with significant differences. On the other hand, Table 5.13 shows the post-hoc analysis when All is chosen for the factor *Profiling Method*. There is no significant difference between CF-IDF and HCF-IDF. Table 5.14 shows the post-hoc analysis of the factor *Publication Content* when CF-IDF is employed. The table indicates that the recommendation strategies with CF-IDF and All significantly outperform those with CF-IDF and Title. Therefore, CF-IDF cannot work when only titles are available. In contrast, the factor *Publication Content* does not influence HCF-IDF and LDA.

Table 5.13: Rankscores and post-hoc analysis for the factor *Profiling Method* at All using Shaffer's MSRB procedure.

**a) Rankscores**

| **Choice** | **M** | **SD** |
|---|---|---|
| CF-IDF | .55 | .33 |
| HCF-IDF | .53 | .32 |
| LDA | .30 | .32 |

**b) Post-hoc analysis p-values**

|  | HCF-IDF | LDA |
|---|---|---|
| CF-IDF | .20 | **.00** |
| HCF-IDF |  | **.00** |

Table 5.14: Rankscores and post-hoc analysis for the factor *Publication Content* at CF-IDF using Shaffer's MSRB procedure.

**a) Rankscores**

| **Choice** | **M** | **SD** |
|---|---|---|
| All | .55 | .33 |
| Title | .40 | .28 |

**b) Post-hoc analysis p-value**

|  | All |
|---|---|
| Title | **.00** |

**The Factor *Temporal Decay Function* × *Publication Content*** Table 5.15 shows the results of the ANOVA regarding the factor *Temporal Decay Function* when a choice of the factor *Publication Content* is fixed and vice versa. According to Table 5.15, there is a significance of the factor *Publication Content*, when Sliding Window is used. The rankscores and post-hoc analysis are shown in Table 5.16; it indicates that All significantly enhances the recommendation performance when Sliding Window is used.

Table 5.15: ANOVA for *Temporal Decay Function* × *Publication Content* interaction.

| **Factor** | **F** | $\eta^2$ | **p** |
|---|---|---|---|
| *Temporal Decay Function* at Title | 0.04 | .00 | .85 |
| *Temporal Decay Function* at All | 3.16 | .03 | .08 |
| *Publication Content* at Sliding Window | 9.44 | .08 | **.00** |
| *Publication Content* at Exponential | 0.56 | .00 | .46 |

Table 5.16: Rankscores and post-hoc analysis for the factor *Publication Content* at Sliding Window using Shaffer's MSRB procedure.

**a) Rankscores**

| **Choice** | **M** | **SD** |
|---|---|---|
| All | .48 | .36 |
| Title | .42 | .32 |

**b) Post-hoc analysis p-value**

| | All |
|---|---|
| Title | **.00** |

### 5.4.3 Influence of Demographic Factors

Mendoza's test is used to examine violation of sphericities with regard to demographic factors including gender, age, highest academic degree, major, years of profession, and current employment type (academia/industry). Subsequently, we conduct a mixed ANOVA with one between-subject factor (i .e., one of the demographic factors) and one within-subject factor (i .e., recommendation strategy), adjusted by Greenhouse-Geisser correction with respect to each demographic factor. The analysis reveals that the demographic factors *Gender* and *Highest Academic Degree* have a significant influence on the recommendation performance. Below, the details of these two factors are described. The details of the other factors whose results are non-significant can be found in Appendix A.

**Gender** Mendoza's test reveals a violation of sphericity in the factor recommendation strategy ($\chi^2(131) = 489.39$, $p = .00$) when comparing male ($n = 96$) and female ($n = 27$) subjects. Table 5.17 shows the result of an ANOVA with a Greenhouse-Geisser correction of $\epsilon = .60$ for the factor recommendation strategy. In Table 5.17, we see a significant difference between subjects grouped by their genders. Table 5.18 shows the post-hoc analysis. We observe that female subjects are more likely to evaluate recommended publications as interesting than male subjects are. However, the factor *Gender* does not make any difference in terms of how each of the twelve recommendation strategies performs compared to the other strategies. In fact, there is no significant difference in the factor *Gender* × *Strategy* in Table 5.17.

**Highest Academic Degree** Referring to the demographic factor *Highest Academic Degree*, we have subjects whose highest academic degree is a Bachelor ($n = 21$), Master ($n = 58$), and PhD ($n = 32$), as well as subjects who are lecturers or professors ($n = 12$). Mendoza's test finds a violation of sphericity in

Table 5.17: Mixed ANOVA with a between-subject factor *Gender* and a within-subject factor *Strategy* with Greenhouse-Geisser correction with F-ratio, effect size $\eta^2$, and p-value.

| **Factor** | **F** | $\eta^2$ | **p** |
|---|---|---|---|
| *Gender* | 9.69 | .08 | **.00** |
| *Strategy* | 16.58 | .14 | **.00** |
| *Gender* × *Strategy* | 1.11 | .01 | .36 |

Table 5.18: Rankscores and post-hoc analysis for the factor *Gender* using Shaffer's MSRB procedure.

**a) Rankscores**

| | **M (SD)** |
|---|---|
| male | .42 (.32) |
| female | .54 (.35) |

**b) Post-hoc analysis p-values**

| | female |
|---|---|
| male | **.00** |

the factor recommendation strategy when comparing the distributions among the factors ($\chi^2(263) = 653.03$, $p = .00$). Table 5.19 shows the result of an ANOVA with a Greenhouse-Geisser correction of $\epsilon = .60$ for the factor recommendation strategy. The analysis reveals a significant difference among subjects grouped by their highest academic degrees. Table 5.20 shows the post-hoc analysis. We observe that subjects whose highest academic degree is a Bachelor are more likely to evaluate recommended publications as interesting than those who are lecturers or professors.

There are significant differences with regard to the demographic factors *Gender* and *Highest Academic Degree*. However, both factors are independent of the recommendation strategies. This indicates that the demographic factors have no influence on which recommendation strategy performs better.

Table 5.19: Mixed ANOVA with a between-subject factor *Highest Academic Degree*, and a within-subject factor, *Strategy* with Greenhouse-Geisser correction, with F-ratio, effect size $\eta^2$, and p-value.

| **Factor** | **F** | $\eta^2$ | **p** |
|---|---|---|---|
| *Highest Academic Degree* | 3.38 | .09 | **.02** |
| *Strategy* | 16.02 | .13 | **.00** |
| *Highest Academic Degree* × *Strategy* | 0.77 | .02 | .75 |

Table 5.20: Rankscores and post-hoc analysis for the factor *Highest Academic Degree* using Shaffer's MSRB procedure.

**a) Rankscores**

| **Degree** | **M (SD)** |
|---|---|
| Bachelor | .53 (.30) |
| Master | .43 (.33) |
| PhD | .44 (.33) |
| lecturer/professor | .32 (.28) |

**b) Post-hoc analysis p-values**

| | Master | PhD | lecturer/professor |
|---|---|---|---|
| Bachelor | .20 | .21 | **.01** |
| Master | ▨ | .72 | .21 |
| PhD | ▨ | ▨ | .09 |

### 5.4.4 Influence of the Amount of Content Available for User Profiles

To investigate the influence of the amount of content available for user profiles, we compute Kendall rank correlation coefficients between the rankscore of each recommendation strategy and the amount of content. As an indicator of the amount of content, we use four measures: the number of tweets, the number of entities, the number of entities per tweet, and the percentage of tweets containing at least one entity. A correlation may show a dependency that could influence the recommendation performance. The results are summarized in Table 5.21. We observe only a few correlations with significant differences. Regarding the number of tweets, a subject has published on average $1,096.82$ (SD: $1,048.46$), as stated in Section 5.3.3. We observe both slight positive and negative correlations with the recommendation strategies using LDA and Exponential. In terms of the number of entities, a subject's tweets contain on average $1,214.93$ entities (SD: $1,181.43$). There is a slight negative correlation with the recommendation strategy LDA × Exponential × All. Referring to the number of entities per tweet, a subject's tweet contains on average 1.07 entities (SD: 0.31). We observe no significant correlation in this regard. Regarding the tweets that contribute to computing user profiles with CF-IDF and HCF-IDF, on average 62.24% of the tweets (SD: 13.55) contain at least one entity. However, there is again no significant correlation. Since we observe no correlation between the amount of content available for user profiles and the novel recommendation strategies (i .e., recommendation strategies using HCF-IDF), we conclude that HCF-IDF is robust against the amount of content.

In fact, the recommender system works for a subject who has published only two tweets.

### 5.4.5 Click Rate on the PDF Files

In the experiment, subjects can click on the titles of the recommended publications to open the corresponding PDF files. On average, subjects click on 4.85 titles (SD: 9.20) of the 60 recommended publications. Thus, the average click rate is 8.08% (SD: 15.33). Table 5.22 shows the click rate per strategy. We run a three-way repeated-measure ANOVA on the rankscores. The results show that the click rates are significantly lower for the recommendation strategies involving LDA compared to CF-IDF and HCF-IDF.

While Table 5.22 shows average click rates of each recommendation strategy, Table 5.23 presents the average precision of clicked PDF files. This precision can be interpreted as the probability that a subject evaluates a recommended publication as "interesting." In Table 5.23, we observe that the values of the strategies using HCF-IDF are high even if recommendations are computed based on only titles.

### 5.4.6 Questionnaire Feedback

At the end of the experiment, the subjects are asked to rate "how easy it was to decide whether a recommended publication is interesting." We use a five-point Likert scale, where values between 1 and 5 indicate very difficult to very easy, respectively. The result is fairly high with an average of 3.68 (SD: 0.88). Regarding the question of "whether the subjects noticed a difference between the twelve strategies," the result is similarly high, with an average of 3.46 (SD: 1.20). In the free comment section, one subject notes that the recommender system failed to pick up his primary field despite his having tweeted about that domain. Apart from this, we receive many positive comments (e .g., interesting, useful). Among them, one subject provides a comment that she would like to use the recommender system in practice.

### 5.4.7 Computation Time

Table 5.24 reports the mean average computation time required to compute recommendations for each strategy. Please note that we report the mean average computation time of 160 subjects despite the total of 123 subjects, since 160 subjects registered for the experiment. We compute recommendations for every

Table 5.21: Correlation coefficients of the rankscore of the twelve recommendation strategies with the amount of content for user profiles. We calculate correlation coefficients using Kendall rank correlation coefficient. In parentheses, p-values are given and marked in bold font, if < .05. Recommendation strategies are sorted by rankscore as shown in Table 5.2.

| | **Recommendation Strategy** | | | **# of tweets** | **# of entities** | **# of entities per tweet** | **Percentage of tweets with entities** |
|---|---|---|---|---|---|---|---|
| | **Profiling Method** | **Temporal Decay Function** | **Publication Content** | | | | |
| 1. | CF-IDF | Sliding Window | All | -.01 (.84) | -.01 (.91) | -.01 (.82) | .01 (.92) |
| 2. | HCF-IDF | Sliding Window | All | -.02 (.77) | .00 (.96) | .05 (.46) | .06 (.32) |
| 3. | HCF-IDF | Sliding Window | Title | -.07 (.26) | -.04 (.51) | .05 (.38) | .06 (.31) |
| 4. | HCF-IDF | Exponential | Title | -.03 (.68) | -.04 (.57) | -.09 (.15) | -.10 (.11) |
| 5. | CF-IDF | Exponential | All | .02 (.80) | .01 (.92) | -.08 (.23) | -.07 (.24) |
| 6. | HCF-IDF | Exponential | All | .02 (.71) | .03 (.68) | -.06 (.32) | -.07 (.24) |
| 7. | CF-IDF | Exponential | Title | .08 (.19) | .07 (.25) | -.05 (40) | -.08 (.20) |
| 8. | CF-IDF | Sliding Window | Title | .09 (.17) | .06 (.32) | -.03 (.65) | -.04 (.50) |
| 9. | LDA | Exponential | Title | .12 (**.05**) | .11 (.07) | .00 (.99) | .00 (.97) |
| 10. | LDA | Sliding Window | Title | .08 (.24) | .06 (.34) | -.01 (.89) | -.01 (.82) |
| 11. | LDA | Exponential | All | -.13 (**.03**) | -.12 (**.05**) | .03 (.65) | .04 (.54) |
| 12. | LDA | Sliding Window | All | -.08 (.25) | -.07 (.28) | .01 (.93) | -.01 (.92) |

Table 5.22: Average click rates on the PDF files. M and SD denote mean and standard deviation, respectively. Recommendation strategies are sorted by rankscore as in Table 5.2.

| | **Recommendation Strategy** | | | **Click Rate (%)** |
|---|---|---|---|---|
| | **Profiling Method** | **Temporal Decay Function** | **Publication Content** | **M (SD)** |
| 1. | CF-IDF | Sliding Window | All | 10.73 (24.73) |
| 2. | HCF-IDF | Sliding Window | All | 10.08 (23.94) |
| 3. | HCF-IDF | Sliding Window | Title | 9.11 (22.21) |
| 4. | HCF-IDF | Exponential | Title | 7.64 (17.28) |
| 5. | CF-IDF | Exponential | All | 9.11 (23.22) |
| 6. | HCF-IDF | Exponential | All | 8.29 (20.31) |
| 7. | CF-IDF | Exponential | Title | 8.94 (20.03) |
| 8. | CF-IDF | Sliding Window | Title | 9.59 (22.81) |
| 9. | LDA | Exponential | Title | 4.23 (13.12) |
| 10. | LDA | Sliding Window | Title | 4.72 (15.38) |
| 11. | LDA | Exponential | All | 9.27 (21.47) |
| 12. | LDA | Sliding Window | All | 5.37 (16.41) |

Table 5.23: Precision of clicked PDF files. Recommendation strategies are sorted by rankscores as shown in Table 5.2.

| | **Recommendation Strategy** | | | **Precision** |
|---|---|---|---|---|
| | **Profiling Method** | **Temporal Decay Function** | **Publication Content** | |
| 1. | CF-IDF | Sliding Window | All | 0.71 |
| 2. | HCF-IDF | Sliding Window | All | 0.65 |
| 3. | HCF-IDF | Sliding Window | Title | 0.55 |
| 4. | HCF-IDF | Exponential | Title | 0.68 |
| 5. | CF-IDF | Exponential | All | 0.71 |
| 6. | HCF-IDF | Exponential | All | 0.61 |
| 7. | CF-IDF | Exponential | Title | 0.47 |
| 8. | CF-IDF | Sliding Window | Title | 0.49 |
| 9. | LDA | Exponential | Title | 0.42 |
| 10. | LDA | Sliding Window | Title | 0.38 |
| 11. | LDA | Exponential | All | 0.44 |
| 12. | LDA | Sliding Window | All | 0.49 |

Table 5.24: Computation time in seconds required by the strategies to calculate recommendations per subject. M and SD denote mean and standard deviation, respectively. Recommendation strategies are sorted by rankscore, as in Table 5.2.

| | **Recommendation Strategy** | | | **Computation time (sec.)** |
|---|---|---|---|---|
| | **Profiling Method** | **Temporal Decay Function** | **Publication Content** | **M (SD)** |
| 1 | CF-IDF | Sliding Window | All | 11.35 (5.36) |
| 2 | HCF-IDF | Sliding Window | All | 17.59 (6.68) |
| 3 | HCF-IDF | Sliding Window | Title | 17.52 (6.68) |
| 4 | HCF-IDF | Exponential | Title | 25.18 (8.14) |
| 5 | CF-IDF | Exponential | All | 14.16 (5.56) |
| 6 | HCF-IDF | Exponential | All | 26.05 (8.31) |
| 7 | CF-IDF | Exponential | Title | 5.15 (4.25) |
| 8 | CF-IDF | Sliding Window | Title | 5.05 (4.23) |
| 9 | LDA | Exponential | Title | 7.50 (5.28) |
| 10 | LDA | Sliding Window | Title | 7.37 (5.28) |
| 11 | LDA | Exponential | All | 361.97 (25.17) |
| 12 | LDA | Sliding Window | All | 361.71 (25.18) |

subject, although some did not start the evaluations. In Table 5.24, standard deviations are high, since computation times highly depends on the number of tweets generated by users. Referring to HCF-IDF, the computation time of the recommendation strategies with All are much longer than those with Title, compared to CF-IDF and LDA. LDA takes a long time, especially when the full texts of scientific publications are used. Please note that we implement the recommendation strategies used in the experiment by ourselves, and they are not optimized.

## 5.5 Discussion

**Summary of main insights** The recommendation strategies with HCF-IDF perform almost equally well compared to the best performing strategy of CF-IDF × Sliding Window × All. There is no significant difference between them, as described in Table 5.3. The strong advantage of HCF-IDF is that it already reaches its peak recommendation performance when only using the titles of the scientific publications. In fact, the post-hoc analysis of the factor *Profiling Method* × *Publication Content* shows that there is no significant difference between the recommendation strategies with Title and those with All when HCF-IDF is employed. This indicates that the recommendation performance of HCF-IDF is similar when using both titles and full texts and when using only titles. The reason for this is that spreading activation over the hierarchical knowledge graph used in HCF-IDF successfully reveals entities that are not explicitly mentioned in titles, but highly relevant to them. Since it is not easy to obtain full texts of scientific publications in reality, for instance due to legal reasons, we believe that this is a highly interesting and promising result. In contrast, CF-IDF works well only when the full texts of the scientific publications are available. In fact, when CF-IDF is employed, the recommendation strategies with All perform significantly better than those using Title. This is because it is difficult to extract enough entities from the titles to construct reasonable publication profiles. Regarding LDA, its recommendation performance is overall low, even if the full texts are available. A possible reason for this is that LDA cannot construct accurate user profiles because social media items are short. Without accurate user profiles, it is impossible to make good recommendations. In fact, a slight correlation between the rankscores of LDA and the number of tweets is observed as reported in Table 5.21. This indicates that subjects who have published more tweets receive better recommendations. Please note that the rankscores are almost the same as precision and nDCG (see Appendix A). Although rankscores are slightly different when using MRR compared to MAP, the order of performance of the twelve recommendation strategies is almost identical. Thus, the findings revealed in the experiment are not influenced by the evaluation metrics.

**Generalizability** We conduct the experiment in the field of economics in a broader sense. The corpus of scientific publications covers the wider field of economics, including social science, political science, and information science. In addition, 31 subjects out of 123 subjects work in domains other than economics, e .g., political science and computer science. To identify whether the recommendation performance is significantly different for subjects from economics and those not in economics, we conduct an ANOVA. The result of the ANOVA shows that majors do not have a significant difference. Thus, we assume that our methods can be transferred to other domains. Furthermore, many domain-specific hierarchical knowledge graphs are freely available in other domains. For instance, MeSH is available for medicine, and ACM CCS for computer science. An overview of freely available hierarchical knowledge graphs[11] is given by W3C. They are of high quality, as they are manually crafted by domain experts. These knowledge graphs are of a similar structure to the STW used in the present experiment. Therefore,

[11] `http://www.w3.org/2001/sw/wiki/SKOS/Datasets`, last accessed on 08/31/2017

HCF-IDF can easily be applied to other fields. Regarding social media platforms, we employ Twitter in the experiment. However, the recommender system can also work with other social media platforms, such as Facebook and LinkedIn. In addition, we observe that the recommendation strategies with HCF-IDF are robust against the number of tweets. In fact, the mean average rankscore of the strategies with HCF-IDF for the subjects whose number of tweets is ranked in the bottom 25% is .55 (SD: .30). It is almost the same as the mean average rankscore for all the subjects, which is .53 (SD: .32).

**Threats to validity** The results of the experiment are potentially influenced by the amount of time that each subject spent completing the evaluations. The subjects spent on average 517.54 seconds (SD: 376.72) to evaluate the 60 recommendations. However, there is no correlation between the rankscore and the amount of time spent completing the experiment. In addition, we randomize the order of the recommendation strategies with respect to each subject, to remove any influence of that order. Another potential threat is that the results may be influenced by how subjects are recruited. One of the subjects notes this in the qualitative feedback. However, we believe that this threat is small. First, there are enough subjects with respect to each demographic factor, as shown in Section 5.3.2. In addition, the same subject mentioned above also states that the method how we acquire our subjects in fact generates a representative sample, as of course economists are the target users of the recommender system. One might be concerned about whether social media items contain substantial information from which to extract users' professional interests. However, the analysis shows that 63% of tweets contain at least one entity of the knowledge graph. Therefore, we assume that it is possible to extract users' professional interests from social media items.

# Chapter 6

# Profiling Data Dynamics on Knowledge Graphs

In Chapters 4 and 5, we demonstrated that profiling methods using knowledge graphs assist in understanding users and generating reasonable recommendations. The knowledge graphs used by the profiling methods are manually maintained and thus of high quality. While the knowledge graphs used in Chapters 4 and 5 rarely change, knowledge graphs such as DBpedia and Wikidata do a lot. Therefore, it is also important to profile the data dynamics of knowledge graphs in order to maintain the integrity of the knowledge graphs. Then, we can use knowledge graphs such as DBpedia and Wikidata for the profiling methods using knowledge graphs. The data dynamics refer to a pattern or process of changes in data, as introduced in Chapter 1. Please note that knowledge graphs following the SKOS specifications used in Chapters 4 and 5 change very little. However, it is crucial to understand the data dynamics of knowledge graphs and maintain their integrity, when using more dynamic knowledge graphs for the profiling methods. Therefore, we profile the data dynamics of knowledge graphs by investigate how the content and structure (i .e., topology) of knowledge graphs influence on the data dynamics of knowledge graphs.

In Section 6.1, we investigate the influence of contents of triples on the data dynamics of triples. Specifically, we apply the linear regression model to predict triples' life spans (i .e., how long a triple is alive) based on its content. Then, in Section 6.2, we explore how topological features of entities, such as node degree, influence on the data dynamics of a knowledge graph. The results contribute to two different applications, which are shown in Chapters 7 and 8.

## 6.1 Profiling Data Dynamics of Triples Using their Content

As stated in Definition 3.4, knowledge graphs are composed of a set of triples. However, only a few studies have investigated the data dynamics of knowledge graphs focusing on triples. For example, Käfer et al. [KAU+13] quantified changes with respect to a set of triples, set of links, and schema signature. They found that most dynamic predicates were about trivial time stamps. In the work of Martin et al. [MUA10], they showed that SPARQL query caching allows to execute queries more efficiently. In addition, Zhang et al. [ZST+15] cached triples that are consumed frequently by SPARQL queries. Thus, it is expected that the profiling result of triple data dynamics can further improve these caching methods. In this vein, this section predicts the data dynamics of triples. Specifically, we predict triples' life spans using a linear regression model. By doing so, we aim to identify which triples are stable and which are ephemeral. The linear regression model applies different features of triples coming from their content: subject pay-level-domain (PLD), predicate, and object form and object PLD. Dividino et al. [DGS15] and Umbrich et al. [UHH+10] attempted to predict data dynamics based on how frequently RDF documents had been changed in the past. In contrast, we profile and predict data dynamics based on the single triples in the RDF documents. The profiling results with two datasets reveal that subject PLD and predicate have a large influence on determining triples' life spans.

In Section 6.1.1, we first introduce the triple features used for the linear regression model. Section 6.1.2 details the regression model that predicts triples' life spans. Subsequently, Section 6.1.3 presents two datasets used to train the model, and Section 6.1.4 describes the resulting model as well as its prediction power.

### 6.1.1 Triple Features

We examine three features: "subject PLD", "predicate", and "object form and PLD". We see these features as contents of triples. We choose these features because Radinsky and Benett [RB13] and Tan and Mitra [TM10] demonstrated that it was possible to predict future changes to web documents by examining their content. Therefore, we assume that it is possible to predict the data dynamics of triples (i .e., triples' life span) by analyzing their "subject PLD", "predicate", and "object form and PLD", which can be considered as contents of triples.

**Subject PLD** Subjects are defined by a URI. From this subject URI, we use the PLD as a feature. For instance, if a subject URI is `http://dbpedia.org/resource/Facebook`, the subject PLD is `http://dbpedia.org`. This feature is motivated by the work of Umbrich et al. [UKL10] that observed entities coming from the same PLD showed similar data dynamics. The PLD of a URI is extracted using Guava[1]. If Guava identifies no PLD, "other subject PLDs" is assigned.

**Predicate** Triples that have a common predicate may demonstrate a similar life span. For instance, a triple whose predicate is `http://dbpedia.org/ontology/areaLand` is assumed to be static, because an area of places such as countries do not change frequently. In contrast, a triple whose predicate is `http://dbpedia.org/ontology/populationTotal` likely disappears and a new triple whose predicate is `http://dbpedia.org/ontology/populationTotal` appears as population statistics are updated. Thus, a triple with the predicate `http://dbpedia.org/ontology/populationTotal` can be assumed to be more ephemeral.

**Object form and PLD** Objects are either a URI or a literal. If an object is defined by a literal, the triple is assigned the feature "literal." Otherwise, it is assigned the PLD of the object URI, as we do for the subject PLD.

### 6.1.2 Prediction of Triple Life Span

We train a linear regression model to predict life spans of triples using the above triple features. We use frequencies of triples over snapshots as life spans of triples. The linear regression model is defined as:

$$LR = z_0 b_0 + z_1 b_1 + z_2 b_2 + \cdots + z_d b_d. \tag{6.1}$$

In Equation 6.1, $z_i$ denotes a coefficient (i .e., weight) of a feature. $z_0$ equals the intercept of the model. $b_i$ is a feature value, and $d$ stands for the number of features in the model. Although $b_0$ is not in the original equation, we introduce it as a constant $b_0 = 1$ to ease the notation. In short, Equation 6.1 is represented as $LR = \boldsymbol{z}^T \boldsymbol{b}$. $\boldsymbol{z}$ and $\boldsymbol{b}$ denote coefficients and feature values, which are $d + 1$ dimensional vectors. Subsequently, we describe a feature value $b_i$ and a feature value vector $\boldsymbol{b}$. Since all the triple features are nominal data, we convert the triples into feature value vectors using one-hot encoding. In the example shown in Table 6.1,

[1] `https://github.com/google/guava/wiki/Release19`, last accessed on 08/31/2017

there are two unique subject PLDs, three unique predicates, and two object forms (i .e., one is a literal and the other is a PLD `db`). This results in an eight-dimensional vector, where the zeroth element is a constant $b_0 = 1$, and the first and second elements show `db` and `uni` (subject PLDs), followed by `db:location`, `db:works`, and `db:population` (predicates), and "literal" and `db` (object form and PLDs). For instance, a triple ⟨`db:Anne_Smith`, `db:location`, `db:Green Village`⟩ is converted into $\{1, 1, 0, 1, 0, 0, 0, 1\}$. The frequency of this triple is 3, since it is available in all three snapshots in Table 6.1. Again, this frequency is used as a life span of the triple.

The coefficients $\boldsymbol{z}$ are learned by the Limited-memory BFGS (Broyden–Fletcher–Goldfarb–Shanno) method [LN89] using the training data. The Limited memory BFGS method is an algorithm for solving unconstrained nonlinear optimization problems. Compared to the stochastic gradient descent method, the Limited-memory BFGS method can reach to the optimal solution with fewer iterations. To avoid overfitting, we use L2 regularization that penalizes models with extreme parameter values. Thus, the optimization function is:

$$\min_{\boldsymbol{z}} \sum_{i=1}^{N} (\boldsymbol{z}^T \boldsymbol{b}_i - y_i)^2 + \lambda \cdot ||\boldsymbol{z}||_2^2, \tag{6.2}$$

where $N$ denotes the number of triples in the training data, $\boldsymbol{b}_i$ stands for a feature value vector of $i$-th triple in the training data, and $y_i$ is the frequency of the $i$-th triple. In summary, the first term shows the residual squared sum (RSS) that is employed as a loss function; the second term is the regularization term that avoids extreme parameter values and mitigates overfitting.

### 6.1.3 Datasets

To train the model and evaluate its prediction power, we use two datasets. Table 6.2 summarizes their descriptive statistics. The datasets are split into training data and test data. We randomly pick 90% of unique triples as training data and the rest as test data in each dataset.

**DyLDO** As the first dataset, we use the Dynamic Linked Data Observatory (DyLDO) dataset[2] [KUH+12]. It has been created to monitor a fixed set of RDF documents on a weekly basis. The dataset is composed of 173 weekly snapshots from 11/27/2012 to 03/27/2016 and covers various well-known data sources as well as less commonly known ones [KUH+12]. The original

[2] `http://swse.deri.org/dyldo/`, last accessed on 08/31/2017

Table 6.1: An example of snapshots over time.

| **$X_{t_1}$: a snapshot at time $t_1$** | | |
|---|---|---|
| db:Anne_Smith | db:location | db:Green_Village |
| db:Anne_Smith | db:works | db:Green_University |
| db:Green_Village | db:population | 224123 |
| **$X_{t_2}$: a snapshot at time $t_2$** | | |
| db:Anne_Smith | db:location | db:Green_Village |
| db:Anne_Smith | db:works | db:Green_University |
| uni:John_Brown | db:location | db:Green_Village |
| uni:John_Brown | db:works | db:Green_Institute |
| db:Green_Village | db:population | 223768 |
| **$X_{t_3}$: a snapshot at time $t_3$** | | |
| db:Anne_Smith | db:location | db:Green_Village |
| db:Anne_Smith | db:works | db:Green_University |
| uni:John_Brown | db:location | db:Green_Village |
| uni:John_Brown | db:works | db:Green_University |
| db:Green_Village | db:population | 223540 |

dataset consists of N-quads $\langle s, p, o, c\rangle$, which correspond to subject, predicate, object, and context. Context is equal to the URI of the RDF document that contains the triple $\langle s, p, o\rangle$. We first remove quads that contain blank nodes from the original dataset, because blank nodes may have different identifiers in different snapshots. Thereafter, we identify RDF documents that have been accessed at every snapshot by analyzing the access logs of the crawler. Then, all unique triples that are contained at one of the identified RDF documents are extracted. In total, the snapshots contain $3,271,944$ unique triples. For each unique triple, we count its frequency, i .e., in how many snapshots it appears. The frequency of a triple is interpreted as its life span. The maximum and minimum frequency are 173 and 1, respectively. On average, each triple is alive in 99.29 snapshots (SD: 77.44) over the entire period. Figure 6.1(a) shows the distribution of triple frequencies. Triples are separated into ephemeral and stable ones. In this thesis, ephemeral triples indicate triples that are deleted shortly after they are created. Thus, they have short life spans. On the other hand, stable triples refer to triples with longer life spans. In terms of triple features, we extract $1,706$ subject PLDs and $3,295$ predicates from all unique triples. In $1,573,797$ (48.10%) triples, the object is defined by a literal. There are $3,059$ object PLDs in triples whose object is a URI. Since we observe many subject PLDs, predicates, and object form and PLDs that are used by only a few triples, we integrate them into one feature each – "other subject PLDs," "other predicates," and "other object PLDs" – to reduce the dimension of feature value vectors. Specifically, we merge subject PLDs, predicates, and object PLDs that are used by 10 or less unique triples into these features. The triple features used by more than 10 triples cover over 99% of unique triples, because the frequencies of subject PLDs, predicates, and object form and PLDs follow the power-law distribution. This power-law distribution is also shown by Tummarello et al. [TDO07]. In result, the number of dimensions of the linear regression model is $d = 2,613$ (i.e., the joint of 705 subject PLDs, $1,335$ predicates, and 573 object forms and PLDs).

**Wikidata** As second dataset, we use Wikidata [VK14], which is one of the largest cross-domain knowledge graphs. We obtain the snapshots from the Wikidata RDF exports[3], where the data are converted into N-triples [EGK+14]. We use 25 snapshots of Wikidata from 04/20/2014 to 08/01/2016. Thus, the snapshots have been captured almost monthly. In total, the dataset contains $73,583,940$ unique triples. The maximum and minimum frequencies of the triples are 25 and 1, respectively. On average, each triple is alive in 16.51 snapshots (SD: 9.14). Figure 6.1(b) shows the distribution of triple frequencies. Regarding the triple features, there is only one unique subject PLD. Thus, the feature of the subject PLD is ignored in the Wikidata dataset. On the other hand, we find $2,204$ predicates. In terms of the objects, $19,291,060$ (26.22%) triples are defined by a literal. There are $239,405$ object PLDs in triples whose objects are defined by a URI. Again, we merge subject PLDs, predicates, and object PLDs that are used by 10 or less unique triples. Similar to the DyLDO dataset, the triple features used by more than 10 triples cover over 99% of unique triples, because the frequencies of predicates and object form and PLDs follow the power-law distribution. All in all, the number of dimensions of the linear regression model is $d = 2,719$ (i.e., $1,739$ predicates and 980 object forms).

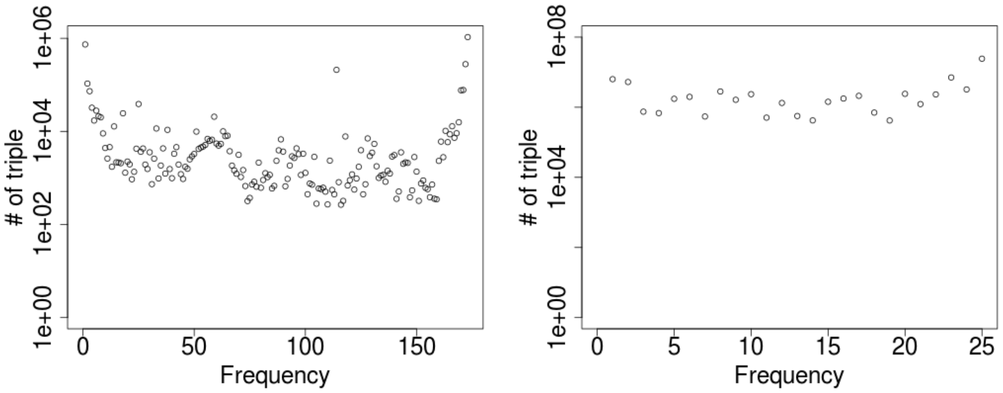
For the linear regression model, we use the implementation provided by GraphLab Create[4]. In addition, we optimize the parameter $\lambda = 316.23$ by 10-fold cross-validation.

[3] `wikidata-simple-statements.nt.gz` from each directory on `https://tools.wmflabs.org/wikidata-exports/rdf/exports/`, last accessed on 11/23/2017

[4] `https://turi.com/learn/userguide/supervised-learning/linear-regression.html`, last accessed on 08/31/2017

Table 6.2: Descriptive statistics of the datasets. The table provides the number of snapshots, the number of unique triples in the entire dataset, and the average frequency of triples. Standard deviation is given in parentheses.

|  | # snapshots | # unique triples | average frequency of triples |
|---|---:|---:|---:|
| DyLDO | 173 | 3,271,944 | 99.29 (77.44) |
| Wikidata | 25 | 73,583,940 | 16.51 (9.14) |



(a) DyLDO dataset

(b) Wikidata dataset

Figure 6.1: Distribution of frequencies of all unique triples in the two datasets.

### 6.1.4 Results

This section first provides the resulting linear regression model. Subsequently, we evaluate the prediction performance of the trained model using the test data.

**Resulting model** The linear regression provides weights (i .e., coefficients) for each triple feature. We start with the resulting model of the DyLDO dataset. In terms of subject PLDs, `ranselrazer.nl`, `fotolog.net`, and `blip.fm` have the largest weights. On the other hand, `today.com` and `nbcnews.com`, which provide news information, have the smallest weights. Referring to predicates, `http://edgarwrap.ontologycentral.com/vocab/edgar\#issued` has the largest weight, while `http://www.w3.org/ns/auth/rsa\#public\_exponent` has the smallest one. The latter is used to note an exponent to encrypt a message. Since such exponents are frequently updated, triples with this predicate are alive only for a short period of time. Regarding object form and PLDs, `rdfabout.com` and `palantir.net` have the largest weights.

Next, we report the resulting model of the Wikidata dataset. Since this dataset only has one subject URI (i .e., `wikidata.org`), subject PLDs are skipped. In terms of predicates, `http://www.wikidata.org/entity/P65c` has the largest weight. It defines a site of astronomical discovery, which is hardly ever changed. In contrast, `https://www.wikidata.org/wiki/Property:P586c` and `https://www.wikidata.org/wiki/Property:P591c` have the smallest weight. These predicates define identifiers of objects, such as plants and enzymes.

**Prediction power** Using the resulting linear regression model, triples' life spans (i .e., frequencies) are predicted using the test data. As evaluation measures, we employ rooted mean squared error (RMSE) and mean absolute error (MAE).

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(y_i - \hat{y}_i)^2}. \tag{6.3}$$

$$MAE = \frac{1}{M}\sum_{i=1}^{M}|y_i - \hat{y}_i|. \tag{6.4}$$

In Equations 6.3 and 6.4, $M$ denotes the number of data points in the test data. $y_i$ is the frequency of the $i$-th triple (i .e., ground truth) and $\hat{y}_i$ is the predicted frequency of the $i$-th triple given by the trained linear regression model. In both measures, lower values indicate better prediction performance. RMSE indicates how well the predicted values fit the linear regression model, and MAE shows how close the predicted values are to the resulting values.

To demonstrate the effectiveness of the linear regression model, we compare the results produced by the mean average of life spans as a baseline. This baseline provides the mean average in the training data to all triples in the test data as a prediction. In addition, we also train the linear regression model using only subject PLDs, predicates, and object form and PLDs, respectively. Thus, we demonstrate which triple feature is most powerful, as well as how well the linear regression model works when all three triple features are used. As shown in Table 6.3, the resulting model outperforms the baseline. The MAE of the model with all triple features is 15.47 in the DyLDO dataset and 3.24 in the Wikidata dataset. This indicates that the model predicts triples' life spans with an error rate of about 10%. Therefore, it is possible to predict the life spans of triples simply by looking at their content, as shown by Radinsky and Bennett [RB13] and Tan and Mitra [TM10]. In addition, the linear regression model with all features outperforms the ones that are solely computed on the features of subject PLD, predicate, and object form and PLD. Thus, all triple features have a positive influence on the prediction of the triples' life spans. Among the three features, subject PLD and predicate have a larger prediction power in the DyLDO dataset. Although the RMSE and MAE of predicates are slightly better than those of subject PLD, we can conclude

Table 6.3: Performance of the prediction of triples' life span. LRM refers to the linear regression model.

| Prediction Model | DyLDO | | Wikidata | |
|---|---|---|---|---|
| | RMSE | MAE | RMSE | MAE |
| Mean average | 77.36 | 73.54 | 8.95 | 7.77 |
| LRM: subject PLD | 44.87 | 27.55 | NA | NA |
| LRM: predicate | 42.72 | 26.39 | 5.72 | 3.82 |
| LRM: object form and PLD | 65.22 | 53.63 | 7.98 | 6.72 |
| LRM: all triple features | 30.77 | 15.47 | 5.16 | 3.24 |

that subject PLD has more power. The reason is as follows: There are fewer features in subject PLD than in predicate, but their prediction performance is competitive. It indicates prediction performance of one subject PLD feature is larger than that of one predicate feature, thus subject PLD is more useful. In the Wikidata dataset, predicate has a larger prediction power than object form and PLD. Moreover, we also investigate topological features such as node degrees of subject URI and object URI, which were used to predict ontology changes in the work of Pesquita and Couto [PC12]. However, the prediction performance using these features is comparable to the baseline. Therefore, we omit the results in this thesis.

## 6.2 Influence of Topological Features of Entities on Data Dynamics

This section analyzes the data dynamics of knowledge graphs with a focus on the topological features of entities. Such features have not been investigated before. In a knowledge graph, entities and relations can be seen as nodes and edges, respectively. We examine how node degrees and node ages influence addition and deletion of edges on knowledge graphs.

Section 6.2.1 introduces a knowledge graph used for this analysis. Section 6.2.2 investigates whether the knowledge graph follows the densification law, which has been observed in different graphs [LKF05]. If the graph follows these patterns, it might be possible to predict new edges of the knowledge graph as social networks do [SCJ12; DTW+12]. Subsequently, we investigate from which kinds of nodes edges are added or deleted in Section 6.2.3, how the destination of added or deleted edges is selected in Section 6.2.4, and how the relation (i .e., predicate) of added or deleted edges is chosen in Section 6.2.5.

### 6.2.1 Dataset

For this investigation, we use 25 snapshots of the Wikidata dataset, which is introduced in Section 6.1.3. We determine changes between two successive points in time by computing the difference between two snapshots. A change is described as an addition or deletion of a triple. We use the notation $(\langle s, p, o\rangle, m, t)$ to represent a change. $\langle s, p, o\rangle$ is a triple and $m$ is a flag that indicates whether a triple is added ($m = 1$) or deleted ($m = -1$). $t$ is the point in time at which the change is made. The set of added and deleted triples produced at a point in time $t$ are extracted from $E_t \setminus E_{t-1}$ and $E_{t-1} \setminus E_t$, respectively. Please note that $E_t$ denotes a set of all triples in a snapshot of a knowledge graph at point in time $t$.

Most of these changes are correct. However, there are incorrect ones because they are made by humans and humans sometimes make mistakes [TAI+14]. To investigate the differences of data dynamics between correct and incorrect changes, we classify changes into correct or incorrect ones. We label a change as incorrect if it is reverted in around four weeks [HPS+15; TAI+14]; otherwise, it is labeled as correct. For example, $(\langle s, p, o\rangle, 1, t)$ is incorrect if $(\langle s, p, o\rangle, -1, t + 3 \text{ weeks})$ is observed. This heuristic was used by Tan et al. [TAI+14] as well as Heindorf et al. [HPS+15]. Although we have 24 successive points in time, we can only label changes made in 23 successive points in time, since we cannot see whether a change is reverted if it was made in the latest successive points in time. In each of the 23 successive points in time, on average $5,357,786.61$ triples are added, of which $333,331.09$ are incorrect. In terms of deleted triples, on average $1,997,224.91$ triples are deleted, of which $177,010.87$ are incorrect. Thus, on average 6.21% of added triples and 8.86% of deleted triples are incorrect.

### 6.2.2 Global Patterns

First, we investigate how the numbers of nodes and edges of the knowledge graph change over time. Figure 6.2 shows the numbers of nodes (i .e., entities and literals) and edges (i .e., triples) over time. We see that both the numbers of nodes as well as edges increase over time.

Then, we investigate whether the knowledge graph follows the densification power law [LKF05], which has been observed in different graphs such as social networks and citation networks. If the graph follows the densification power law, it has the relation $|E_t| \propto |Q_t \cup L_t|^{\alpha}$, where $\alpha$ is an exponent that is $\alpha \in [1, 2]$. Please note that $Q_t$ and $L_t$ denote a set of all entities at a point in time $t$ and a set of all literals at a point in time $t$, respectively. Figure 6.3 (a) represents

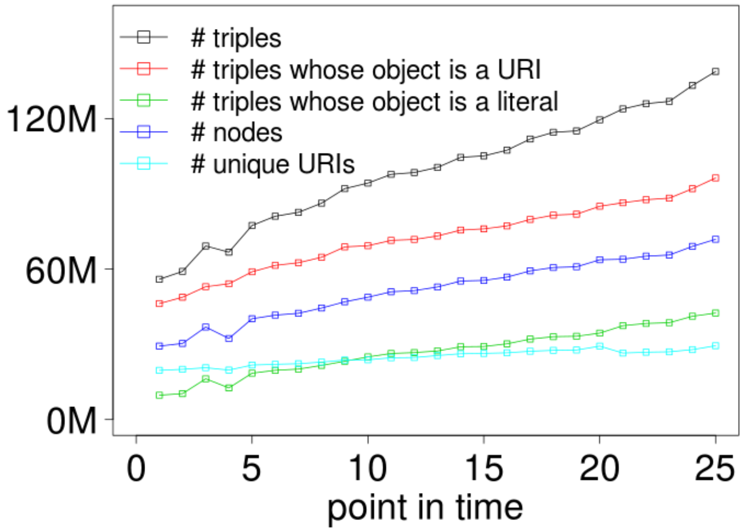
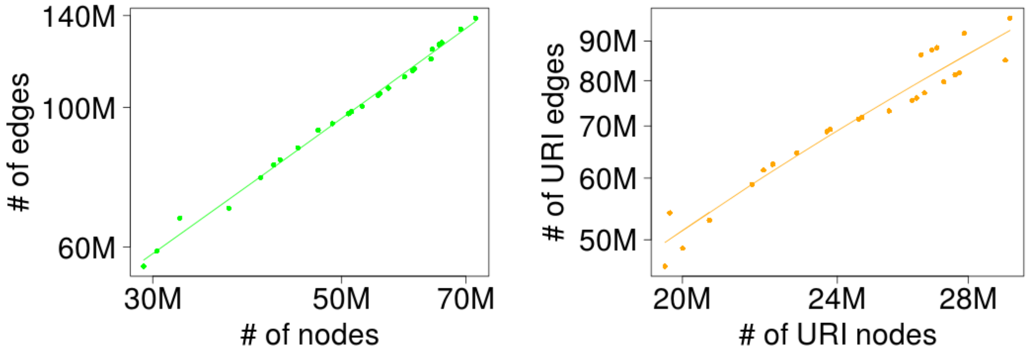


Figure 6.2: The number of nodes (i .e., entities and literals) and edges (i .e., triples) over time.

the relation between the number of nodes $|Q_t \cup L_t|$ and the number of edges $|E_t|$. Please note that both axes are in logarithmic scale. The plots fit well into a line, but do not follow the densification power law, since the exponent is $\alpha = 0.97$. The reason for this is that a literal can only inherently hold one in-degree (i .e., one edge). Thus, the graph becomes increasingly sparse as the number of literals increases.

Please note we treat every literal as one node despite two other options: (a) treating lexically identical literals as one node and (b) treating literals that are lexically identical and used by a same predicate as one node. The latter case is motivated by the idea that literals have different semantics depending on the contexts in which they are used. Thus, our results could be biased towards this decision. However, most literals have actually only one incoming edge in both cases. We have investigated this by computing the between the number of nodes in our setting with the number of nodes in the cases (a) and (b) as follows: $\frac{\text{the number of nodes in the case (a)}}{thenumberofnodesinthisthesis} = 0.85$ and $\frac{\text{the number of nodes in the case (b)}}{thenumberofnodesinthisthesis} = 0.96$. Since the difference with the two cases is low, the influence by how we treat literals is small.

We further investigate the densification power law by excluding the influence of the literal nodes. To this end, we examine the relation $|\langle s, p, o\rangle \in E_t : o \in Q_t| \propto |Q_t|$ as shown in Figure 6.3 (b). $|\langle s, p, o\rangle \in E_t : o \in Q_t|$ denotes the number of edges whose objects are a URI at a point in time $t$. In this case, the exponent is $\alpha = 1.56$, thus following the densification power law. Therefore, we conclude



(a) The number of edges $|E_t|$ versus the number of nodes $|Q_t \cup L_t|$

(b) The number of edges whose objects are a URI $|\langle s, p, o\rangle \in E_t : o \in Q_t|$ versus the number of URI nodes $|Q_t|$

Figure 6.3: The number of edges versus the number of nodes. Both axes are in logarithmic scale.

that the connection among entities (i.e., a URI node) on the knowledge graph becomes increasingly dense over time, thus following other graphs [LKF05].

### 6.2.3 Edge Initiation

In this section, we investigate by which kinds of nodes edges are added and deleted. In particular, we examine this from the topological features node degree, node age, and the last point in time at which a node was edited.

**Node degree** We first explore the influence of a node degree on edge addition and edge deletion. Do rich nodes (i.e., nodes with a high degree) bring more triples to knowledge graphs? For the assessment, we compute the in-degree as well as the out-degree of subject nodes of added and deleted edges. To this end, following the definition from Leskovec et al. [LBK+08], we compute the average number of edges added or deleted by a node of a degree $d$ as:

$$ed(d, m) = \sum_{t \in T} \frac{|\{(\langle s, p, o\rangle, m, t) : deg_{t-1}(s) = d\}|}{|\{x : deg_{t-1}(x) = d\}|}, \tag{6.5}$$

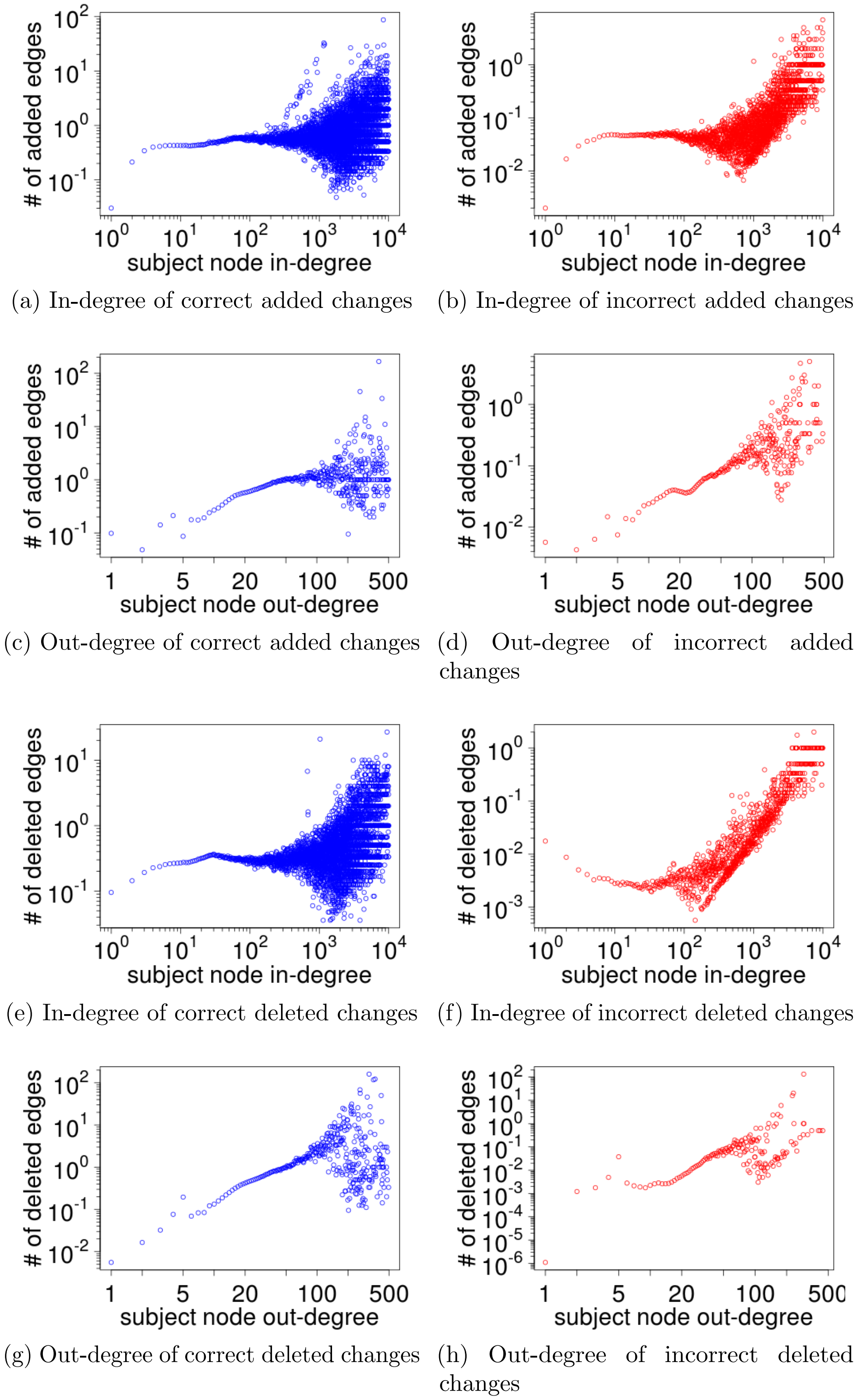
where $m$ is the flag indicating addition or deletion; $x$ is an arbitrary node on the knowledge graph; and $deg(x)$ stands for the degree (either in-degree or out-degree) of a node $x$. Thus, the numerator indicates the number of added or deleted edges between $t-1$ and $t$, whose degree of a subject is $d$. $ed(d, m)$ is normalized by the number of nodes of degree $d$ that exist just before this step. Figure 6.4 illustrates both in-degree and out-degree of subject nodes of added and deleted edges with respect to correct and incorrect changes. Please note that both axes are in logarithmic scale. In Figures 6.4 (a) and (e), we observe that the number of correct added and deleted edges starts increasing after the in-degree reaches 1000. Similarly, Figures 6.4 (b) and (f) show that the number of incorrect added and deleted edges starts increasing after the in-degree reaches 100. Regarding the out-degree of subject nodes, Figures 6.4 (c), (d), (g) and (g) indicate that subject nodes with larger out-degree more likely generate both correct and incorrect changes.

**Node age** We examine the influence of node age on edge addition and edge deletion. To this end, we compute $ed(a, m)$, the average number of edges added or deleted by nodes of age $a$, as follows:

$$ed(a, m) = \sum_{t \in T} \frac{|\{(\langle s, p, o\rangle, m, t) : t - t_c(s) = a\}|}{|\{x : t - t_c(x) = a\}|}. \tag{6.6}$$

The numerator counts the number of added or deleted edges where the age of the subject is $a$. $t_c(s)$ returns a point in time at which a subject node was generated. The number is normalized by the number of nodes whose age is $a$. Please note that, to avoid truncation effects, we remove the nodes that appear in the first snapshot of this analysis. We can see only that these nodes were generated between 10/30/2012 (i .e., the launch of Wikidata) and 04/20/2014 (the first snapshot of Wikidata). Thus, their actual ages may vary too much. Figure 6.5 plots the average number of added and deleted edges by a subject node whose age equals $a$. Please note that the age is represented on the x-axis by the number of points in time in Figure 6.5. Since the period between two successive points in time is approximately 36.05 days, nodes whose $a = 3$ are 108.14 days old. As observed in the study of Leskovec et al. [LBK+08] as well, there is a small spike at $a = 0$ in Figure 6.5 (a). The spike corresponds to nodes that generate edges at the initial stage but never add further edges to the knowledge graph. In addition, we observe that the average number of added edges slightly decreases as subject nodes become older. In terms of deleted edges, Figure 6.5 (b) shows that the number of deleted edges decreases as subject nodes get old. This indicates that the older subject nodes are likely to be abandoned (i .e., to no longer be edited). Finally, we do not observe a large difference between the curves of correct and incorrect changes.

**Node last edit** In addition to node ages, we also investigate the influence of the period of time since the node was last edited. Do nodes that were edited



(a) In-degree of correct added changes

(b) In-degree of incorrect added changes

(c) Out-degree of correct added changes

(d) Out-degree of incorrect added changes

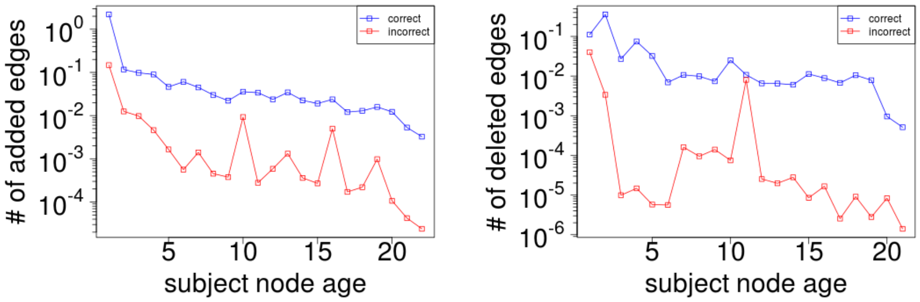
(e) In-degree of correct deleted changes

(f) In-degree of incorrect deleted changes

(g) Out-degree of correct deleted changes

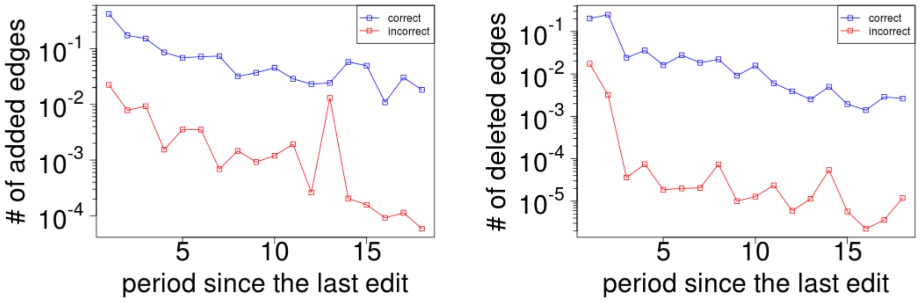
(h) Out-degree of incorrect deleted changes

Figure 6.4: The average degree of subject nodes of added and deleted edges. The x-axis shows the average degree of nodes, and the y-axis indicates the number of added or deleted edges. Both axes are in logarithmic scale.



(a) Added edges (b) Deleted edges

Figure 6.5: The average number of added and deleted edges with a subject node of age $a$. The y-axis is in logarithmic scale.



(a) Added edges (b) Deleted edges

Figure 6.6: The average number of added and deleted edges with a subject node that was last edited $b$ points in time ago. The y-axis is in logarithmic scale.

recently add or delete more edges? We compute $ed(b, m)$, the average number of edges added or deleted in the period $b$, as defined in Equation 6.7:

$$ed(b, m) = \sum_{t \in T} \frac{|\{(\langle s, p, o \rangle, m, t) : t - tl(s) = b\}|}{|\{x : t - tl(x) = b\}|}, \tag{6.7}$$

where $tl(s)$ refers to the point in time at which a node $s$ was last edited. The numerator counts the number of edges that are added or deleted by a node that was last edited $b$ points in time ago. Then, it is normalized by the number of nodes that were last edited $b$ points in time ago. Figure 6.6 illustrates the results. Similar to the node age, the numbers of added and deleted edges are decreasing over time. In addition, the numbers of correct and incorrect changes are decreasing as well. These results indicate that nodes will not be edited if they are abandoned for a longer time.

### 6.2.4 Edge Destination Selection

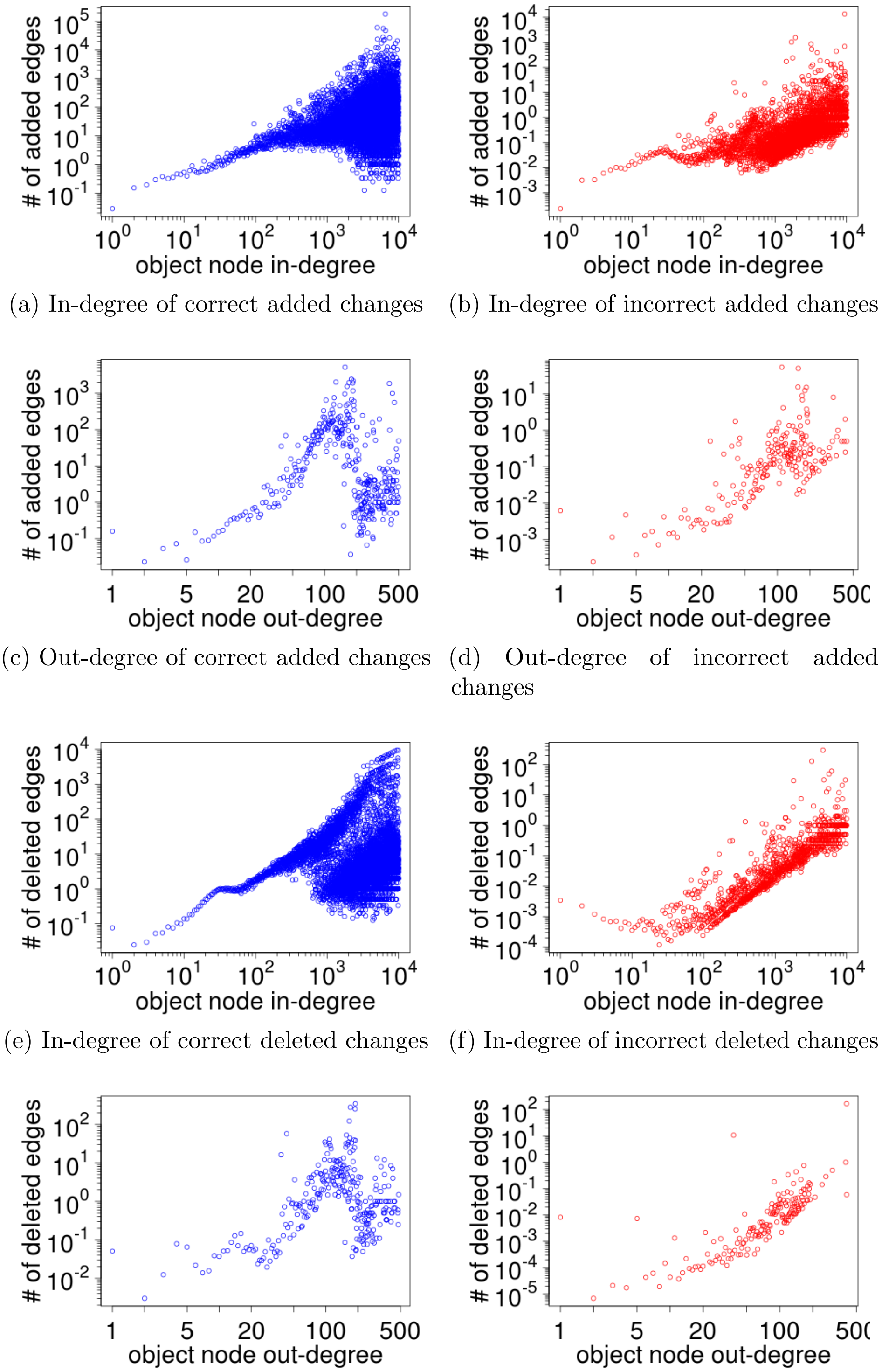
In this section, we examine how the edge destination (i .e., object) of added and deleted edges is selected. Again, we use topological features such as node degree, node age, and node last edit.

**Node degree** We investigate the influence of node degree on edge destination selection in the knowledge graph. The preferential attachment model [BA99] is known and observed in different graphs [LBK+08]. In the preferential attachment model, the likelihood of receiving new edges increases with the node degree. Do knowledge graphs also follow this model? To examine this, we compute $ed(d, m)$, the average number of added and deleted edges with respect to different object degrees, as follows:

$$ed(d, m) = \sum_{t \in T} \frac{|\{(\langle s, p, o \rangle, m, t) : d_{t-1}(o) = d\}|}{|\{x : d_{t-1}(x) = d\}|}. \tag{6.8}$$

The numerator is the number of added or deleted edges between $t-1$ and $t$ whose degree of an object is $d$. It is normalized by the number of nodes of degree $d$ that exist just before this step. Figure 6.7 presents the results. Please note that both axes are in logarithmic scale. As shown by Leskovec et al. [LBK+08], if a graph evolves randomly such as the Erdős-Rényi random model, the line will be flat because the destination node (i .e., object) is chosen independently of its degree. In contrast, in Figures 6.7 (a) and (b), we observe that the knowledge graph follows the preferential attachment model in terms of both correct and incorrect changes. In addition, we also observe that the knowledge graph follows this model in the deleted changes as shown in Figures 6.7 (e) and (f). In the in-degree of both added and deleted changes, the incorrect changes fit the relation $ed(d, m) \propto d^{\alpha}$ better, since the distribution of the number of added or deleted edges with high degrees is narrow. Regarding out-degree, we observe that the number of added and deleted edges follows the relation $ed(d, m) \propto d^{\alpha}$ until the out-degree reaches 100 as shown in Figures 6.7 (c) and (g). When the out-degree is over 100, the number of added and deleted edges decreases. In contrast, we do not see this trend for incorrect changes as shown in Figures 6.7 (d) and (h).

**Node age** We examine the influence of age of object nodes on addition and deletion of edges in the knowledge graph. Do older nodes receive more edges, since they are more experienced and known? We compute the average number of



(a) In-degree of correct added changes

(b) In-degree of incorrect added changes

(c) Out-degree of correct added changes

(d) Out-degree of incorrect added changes

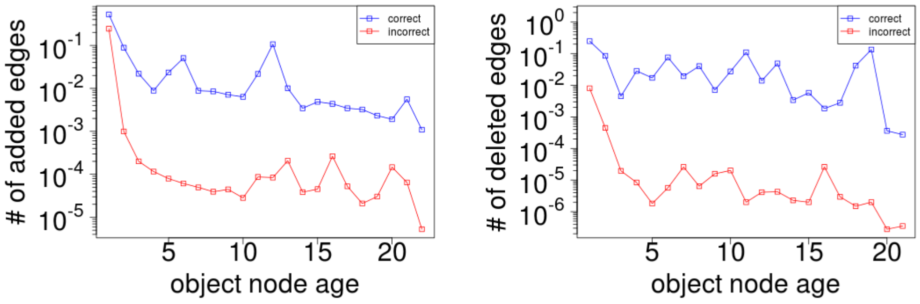
(e) In-degree of correct deleted changes

(f) In-degree of incorrect deleted changes

(g) Out-degree of correct deleted changes

(h) Out-degree of incorrect deleted changes

Figure 6.7: The average degree of object nodes of added and deleted edges. The x-axis shows the average degree of nodes and the y-axis indicates the number of added or deleted edges. Both axes are in logarithmic scale.



(a) Added edges

(b) Deleted edges

Figure 6.8: The average number of added and deleted edges with an object node of age $a$. The y-axis is in logarithmic scale.

edges added or deleted by nodes of age $a$ as follows:

$$ed(a, m) = \sum_{t \in T} \frac{|\{(\langle s, p, o\rangle, m, t) : t - t_c(o) = a\}|}{|\{x : t - t_c(x) = a\}|}. \tag{6.9}$$

Again, we remove the nodes that appear in the first snapshot from the analysis, as we do in Section 6.2.3. Figure 6.8 plots the average number of added and deleted edges whose objects are a node of age $a$. Similar to Figure 6.5, new nodes receive more edges. Regarding the correctness of changes, over 94% of added changes are correct at each age except when $a = 0$, when only 69.03% are correct. Thus, newer nodes more frequently receive incorrect changes. In addition, the probability of incorrect changes is also relatively high at $a = 0$ for the deleted edges.

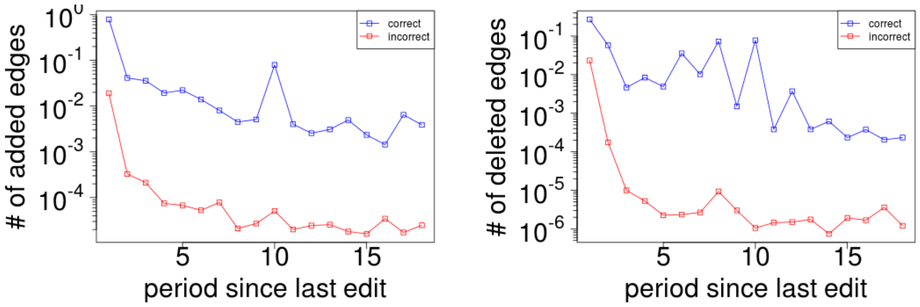
**Node last edit** Again, we examine the influence of the period since the object node was edited. We compute the average number of added and deleted edges with an object node that was edited $b$ points in time ago, as follows:

$$ed(b, m) = \sum_{t \in T} \frac{|\{(\langle s, p, o\rangle, m, t) : t - tl(o) = b\}|}{|\{x : t - tl(x) = b\}|}. \tag{6.10}$$

Figure 6.9 illustrates the result. We again observe that both added and deleted edges decrease. This indicates that the nodes will not be used as destination if they are abandoned for a longer time.

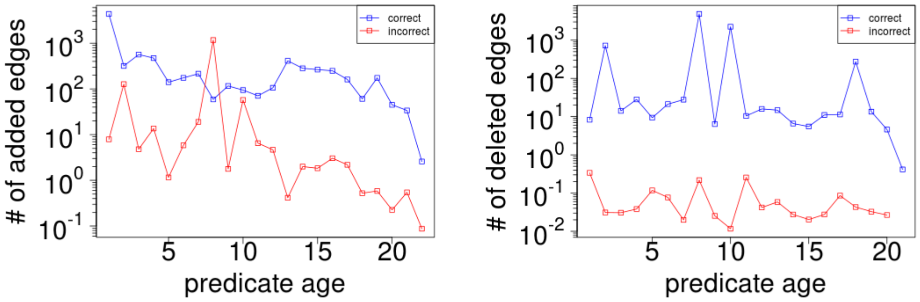
### 6.2.5 Relation Selection

Compared to simple graphs, edges of knowledge graphs indicate different relations (i .e., `isMarriedTo`). Therefore, in this section, we analyze the influence of different relations on the data dynamics of the knowledge graph.



(a) Added edges (b) Deleted edges

Figure 6.9: The average number of added and deleted edges with an object node that was last edited $b$ points in time ago. The y-axis is in logarithmic scale.



(a) Added edges (b) Deleted edges

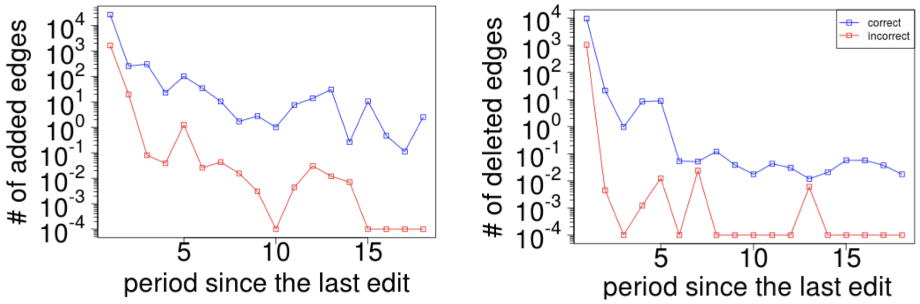
Figure 6.10: The average number of added and deleted edges with a relation of age $a$. The y-axis is in logarithmic scale.

**Relation age** As we did in Sections 6.2.3 and 6.2.4, we examine the influence of a relation (i .e., predicate) age. We compute $ed(a, m)$, the average number of added and deleted edges with a relation of age $a$, as:

$$ed(a, m) = \sum_{t \in T} \frac{|\{(\langle s, p, o\rangle, m, t) : t - t_c(p) = a\}|}{|\{x : t - t_c(x) = a\}|}. \tag{6.11}$$

The numerator counts the number of added or deleted edges with a relation whose age is $a$, normalized by the number of relations with that age. We remove the relations that appear in the first snapshot to avoid truncation effects. Figure 6.10 illustrates the results. We observe several peaks in both added and deleted edges. In Figure 6.10 (a), we observe that added edges decrease as relations get older. On the other hand, the deleted edges do not decrease as shown in Figure 6.10 (b).

**Relation last edit** Furthermore, we examine the influence of the period since a relation was last used. We compute the average number of added and deleted



(a) Added edges (b) Deleted edges

Figure 6.11: The average number of added and deleted edges with a relation that is last used $b$ points in time ago. The y-axis is in logarithmic scale.

edges with a relation that was last used $b$ points in time ago as follows:

$$ed(b,m) = \sum_{t \in T} \frac{|\{(\langle s,p,o \rangle, m, t) : t - tl(p) = b\}|}{|\{\langle s,p,o \rangle : t - tl(p) = b\}|}. \tag{6.12}$$

The numerator indicates the number of added or deleted edges whose predicate (i .e., relation) was last used $b$ points in time ago. The denominator denotes the number of triples, whose predicate was last used $b$ points in time ago. Figure 6.11 plots the average number of added and deleted edges with respect to a relation that was used $b$ points in time ago. We again observe that both added and deleted edges decrease. In addition, Figure 6.11 indicates that incorrect changes likely utilize a relation that was recently used.

# Chapter 7

# Application III: Crawling Strategy

Many applications that use data from knowledge graphs have been developed and used. The data from knowledge graphs are available as RDF documents on the web. Applications that use these data often pre-fetch RDF documents and store them as local copies, or build an index of them to accelerate access and search. However, Chapter 6 and recent investigations [KAU+13; DSG+13; DGS+14] showed that data from knowledge graphs are dynamic and subject to changes. Thus, the local copies or indices do not always reflect the current state of the data and need to be updated. In fact, Gottron and Gottron [GG14] observed that the accuracy of indices built over RDF documents dropped by 50% after as few as 10 weekss (except schema-level ones). Hence, it is necessary for the applications to cope with constant data updates to guarantee the quality of service. Ideally, the local copies would be kept up-to-date by continuous visits to all RDF documents. However, in the real world, LOD applications have to consider limitations of computational resources such as bandwidth and computation time. Due to these limitations, we have to build an efficient crawling strategy to update local copies of RDF documents.

Therefore, we propose a novel crawling strategy for RDF documents based on triples' life spans, which are predicted based on the linear regression model shown in Section 6.1. We assume that predicting the data dynamics of RDF documents on the level of the atomic units (i .e., triples) provides more fine-grained insights and enables a better prediction of the data dynamics.

We first present the problem statement in Section 7.1. Subsequently, the novel crawling strategy is presented in Section 7.2. Section 7.3 details the experiment of the crawling strategy and Section 7.4 reports the results of the experiment. Finally, we discuss the results in Section 7.5.

## 7.1 Problem Statement

We develop a crawling strategy to keep local copies of RDF documents up-to-date. $c$ refers to a context, i .e., a URI of an RDF document, and $C = \{c_1, c_2, \ldots, c_m\}$ is a set of target RDF documents that are stored as local copies and need to be updated. The crawling strategy computes a preference score $ps(c, t)$ of each RDF document $c \in C$ at each point in time $t$. It preferentially crawls RDF documents whose preference scores $ps(c, t)$ are higher and updates their local copies. Crawling is stopped when the bandwidth reaches its limit $\kappa$. The limit $\kappa$ is a maximum number of triples which is calculated as the sum of triples obtained from the visited RDF documents. Then, the preference scores are updated and crawling is restarted at the subsequent point in time. RDF documents contains triples $\langle s, p, o\rangle$, where $s$, $p$, and $o$ correspond to the subject, predicate, and object. The data of the RDF document $c$ at a point in time $t$ is referred by $X_{c,t}$ (i .e., the set of triples in the RDF document $c$ at a point in time $t$). Furthermore, $|X_{c,t}|$ indicates the number of triples in the RDF document $c$ at a point in time $t$. We define the overall data including all target RDF documents as $X_t = \bigcup_{c \in C} X_{c,t}$ and the series of data as $X = \{X_{t_1}, X_{t_2}, \ldots, X_{t_n}\}$. In contrast, $X'$ refers to the data of the local copies.

## 7.2 Crawling Strategy Based on Triple Life Span

We describe a novel crawling strategy based on the linear regression model shown in Section 6.1. The model predicts triples' life spans. The crawling strategy provides a preference score to an RDF document as follows:

$$ps_{lr}(c, t) = (\frac{1}{|X_{c,t}|} \sum_{\langle s,p,o\rangle \in X_{c,t}} LR(\langle s, p, o\rangle))^{-1} \tag{7.1}$$

The function $LR(\langle s, p, o\rangle)$ returns a triple's life span predicted by the linear regression model for a given triple $\langle s, p, o\rangle$. We compute the mean average of triples' life spans by averaging over $LR(\langle s, p, o\rangle)$ for all triples in the RDF document. Finally, a preference score is defined by the reciprocal of the mean average. We take the reciprocal for the following reason. As stated in the problem statement in Section 7.1, crawling strategies visit RDF documents starting from those with larger preference scores. However, the RDF documents with smaller triples' life spans should be visited preferentially, since they contain more ephemeral and dynamic triples. Therefore, we take the reciprocal as output to reverse the order of the RDF documents.

## 7.3 Experiment

We conduct an experiment to evaluate the performance of the novel crawling strategy. We first introduce the baseline of the crawling strategy in Section 7.3.1. Subsequently, Section 7.3.2 describes the two datasets used in the experiment. Thereafter, Section 7.3.3 presents the setups of the experiment. Finally, we introduce two metrics that are used to evaluate crawling strategies in Section 7.3.4.

### 7.3.1 Baseline

Dividino et al. [DGS15] developed a novel crawling strategy based on LOD source dynamics and reported that it performed best compared to other strategies. In their work, LOD source refers to a set of RDF documents from a same PLD [DGS15]. In contrast, we conduct crawling with respect to RDF documents. We do this because we believe that crawling with respect to RDF documents is more common. Dividino et al. [DGS15] compared their crawling strategy to those based on LOD sources' age, PageRank [BP98; PBM+99], size, amount of change between the last two observations, and change rate between the last two observations. Since the authors reported that their crawling strategy based on LOD source dynamics outperformed the others, we employ it as a baseline in our experiment. The crawling strategy assigns preference scores considering history in terms of how many triples in each RDF document have been updated in the past. The preference score is computed using the equation below:

$$ps_{dynamics}(c, t_i) = \sum_{i=t_1}^{t_i} \frac{\delta(X_{c,t_{lu(c,lu(c,i)-1)}}, X_{c,t_{lu(c,i)}})}{t_{lu(c,i)} - t_{lu(c,lu(c,i)-1)}}. \tag{7.2}$$

$lu(c, i)$ is a function that returns the latest point in time at which the given RDF document $c$ was crawled at point in time $i$. Thus, $lu(c, i) \leq i$. This function can be used recursively. For example, the update prior to the last update is represented as $t_{lu(c,lu(c,i)-1)}$. $\delta$ is a function that returns the degree of difference

Table 7.1: Descriptive statistics of the datasets. The table provides the number of snapshots, the number of RDF documents, and the average number of triples per snapshot. Standard deviation is given in parentheses.

| | # snapshots | # RDF documents | average # triples per snapshot |
|---|---:|---:|---|
| DyLDO | 173 | 11,917 | 1,877,875.82 (76,203.44) |
| Wikidata | 25 | 9,753,532 | 48,609,241.20 (7,500,579.23) |

between two RDF documents (two LOD sources in their original work). We use $\delta$ based on the Jaccard distance defined as:

$$\delta(X_{c,t_1}, X_{c,t_2}) = 1 - \frac{|X_{c,t_1} \cap X_{c,t_2}|}{|X_{c,t_1} \cup X_{c,t_2}|}. \tag{7.3}$$

We use it, since Dividino et al. [DGS15] reported $\delta$ based on the Jaccard distance performs slightly better than $\delta$ based on the Dice coefficient.

### 7.3.2 Datasets

The experiment uses the two datasets that are introduced in Section 6.1.3. Table 7.1 provides the descriptive statistics of these datasets.

**DyLDO** As the first dataset, we use the DyLDO dataset, which is introduced in Section 6.1.3. From the original dataset, we first remove quads that contain blank nodes, because these nodes may have different identifiers in different snapshots in this dataset. Thereafter, we identify RDF documents that have been crawled in every snapshot by analyzing the access logs. The identified RDF documents are the target of the crawling strategies. As result, the dataset contains a total of $11,917$ RDF documents.

**Wikidata** As the second dataset, we use Wikidata dataset, which is introduced in Section 6.1.3. The original dataset consists of only triples. Thus, we consider triples that share a common subject URI as one RDF document. Then, we first extract triples whose subject URI appears in all the snapshots. As result, we find $9,753,532$ subject URIs that are treated as RDF documents.

### 7.3.3 Setups

Following Dividino et al. [DGS15], the experiment compares the crawling strategies in two setups: single-step and iterative progression. Furthermore, we simulate different bandwidth constraints along with Dividino et al. [DGS15]. The two individual experiments are described below.

**Single-step** We evaluate the performance of the crawling strategies for a single update of a local copy. We start from a perfect copy at a point in time $t_i$ and compare the quality of the local copy at a point in time $t_{i+1}$ achieved by different crawling strategies.

**Iterative progression** We evaluate how the quality of the local copies change over a longer period of time when considering iterative updates. Starting from a perfect copy at a point in time $t_i$, we aim to measure how well different crawling strategies perform in terms of maintaining an accurate local copy at subsequent points in time $t_{i+1}, t_{i+2} \ldots t_{i+n}$. We evaluate local copies at up to $n = 20$ subsequent points in time (approximately 5 months) for the DyLDO dataset, and $n = 4$ points in time (approximately 4 months) for the Wikidata dataset.

In both crawling strategies and both setups, we compute preference scores based on available history information. The history is composed of the last 50 snapshots for the DyLDO dataset, and 8 snapshots for the Wikidata dataset. Therefore, we experiment starting from $t =$ `2013-11-24` for the DyLDO dataset and $t =$ `2015-05-11` for the Wikidata dataset. In the single-step setup, we slide the starting point $t_i$ by one point in time. For the iterative progression setup, we slide the starting point $t_i$ by the step of 10 points in time for the DyLDO dataset and by the step of 2 points in time for the Wikidata dataset. Referring to the baseline, we compute the preference scores at a point in time $t_i$, examining the last 50 snapshots for the DyLDO dataset and 8 snapshots for the Wikidata dataset. Again, the initial history information is the same in both setups. The preference scores of the RDF documents are continuously updated in the iterative progression setup. Thus, the size of the history information increases along with the iterations in the iterative progression setup. In terms of the linear regression model, we train the model over the first 50 snapshots for the DyLDO dataset, and first 8 snapshots for the Wikidata dataset. In contrast to the baseline, we use the same linear regression models at all points in time and do not update it. We do this to demonstrate its generalizability over time. To construct the linear regression model, we first extract all unique triples in the first 50 snapshots and 8 snapshots of the respective datasets. Then, we count the frequency of subject PLDs and predicates, and object PLDs. Thereafter, we take into account only the features that are used by more than 10 unique triples, and integrate all of the others.

Referring to bandwidth constraint, we increase the relative bandwidth stepwise from 0% to 5% in intervals of 1%, from 5% to 20% in intervals of 5%, and from 20% to 100% in intervals of 20% of all available triples in each snapshot. Therefore, bandwidth at each point in time $t_i$ is calculated as $\kappa =$ (relative bandwidth)$\cdot|X_{t_{i+1}}|$. We compute $\kappa$ using the number of triples in the snapshot at the next point in time $|X_{t_{i+1}}|$. In fact, $|X_{t_{i+1}}|$ is not known at $t_i$, but we use it to ensure precision and recall at 1.0. If we calculate $\kappa$ based on the size of the snapshot at $t_i$ and $|X_{t_{i+1}}| > |X_{t_i}|$, the precision and recall might not reach 1.0 when the relative bandwidth is 100% since the strategy cannot visit and crawl all RDF documents. We believe that the influence of this network resource computation is low, since the size of snapshots does not vary greatly over time.

### 7.3.4 Metrics

We evaluate the resulting local copies using precision and recall, which are defined in Equations 7.4 and 7.5, respectively.

$$precision(X'_t, X_t) = \frac{|X'_t \cap X_t|}{|X'_t|} \tag{7.4}$$

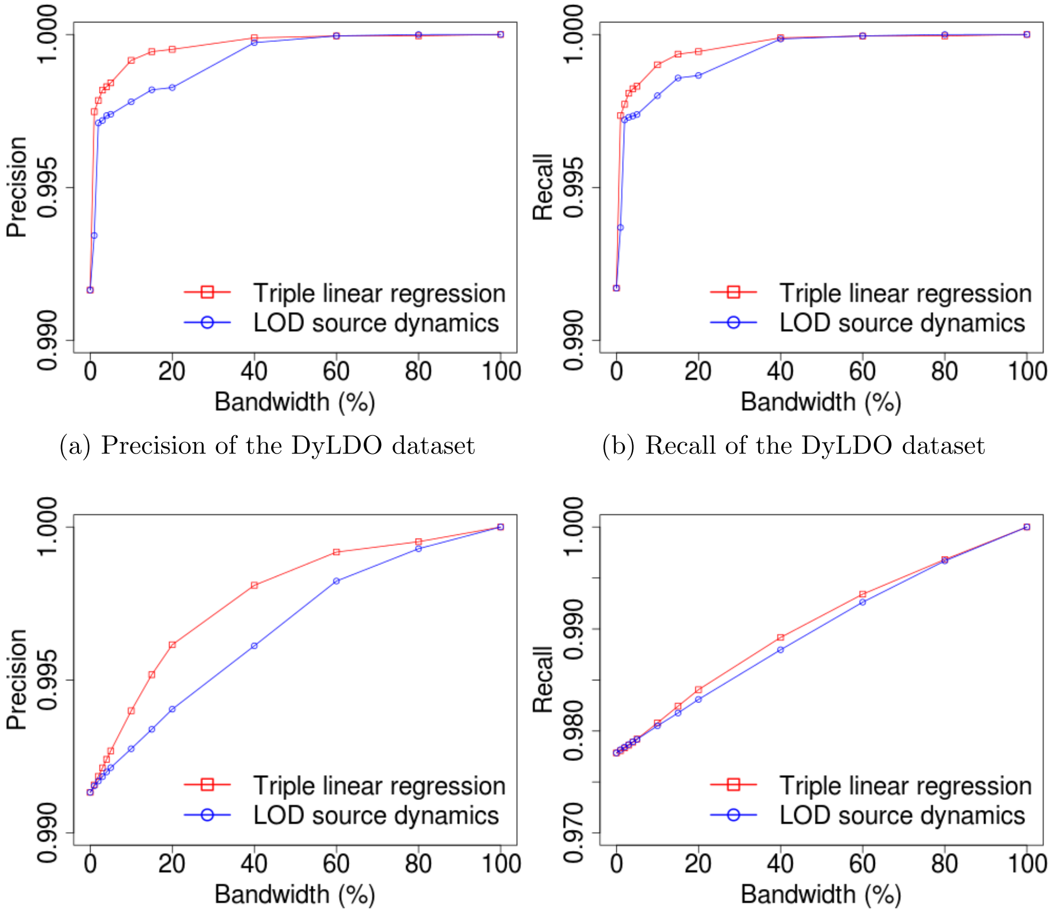
$$recall(X'_t, X_t) = \frac{|X'_t \cap X_t|}{|X_t|} \tag{7.5}$$

In both equations, $X'_t$ denotes the resulting local copy of the RDF documents at a point in time $t$. $X_t$ is the data of all target RDF documents on the web, i .e., a perfect up-to-date local copy at a point in time $t$, which is considered as ground truth.

## 7.4 Results

We report the results of the experiment with respect to the two setups.

**Single-step** Figure 7.1 shows the precision and recall of the local copies produced by the single-step setup when varying the relative bandwidth. Overall, the novel strategy outperforms the baseline in terms of both precision and recall. Regarding precision, when the relative bandwidth is small (< 5%), the difference between the two strategies is small. However, the difference becomes larger as the relative bandwidth increases. Finally, the difference between the crawling strategies



(a) Precision of the DyLDO dataset

(b) Recall of the DyLDO dataset

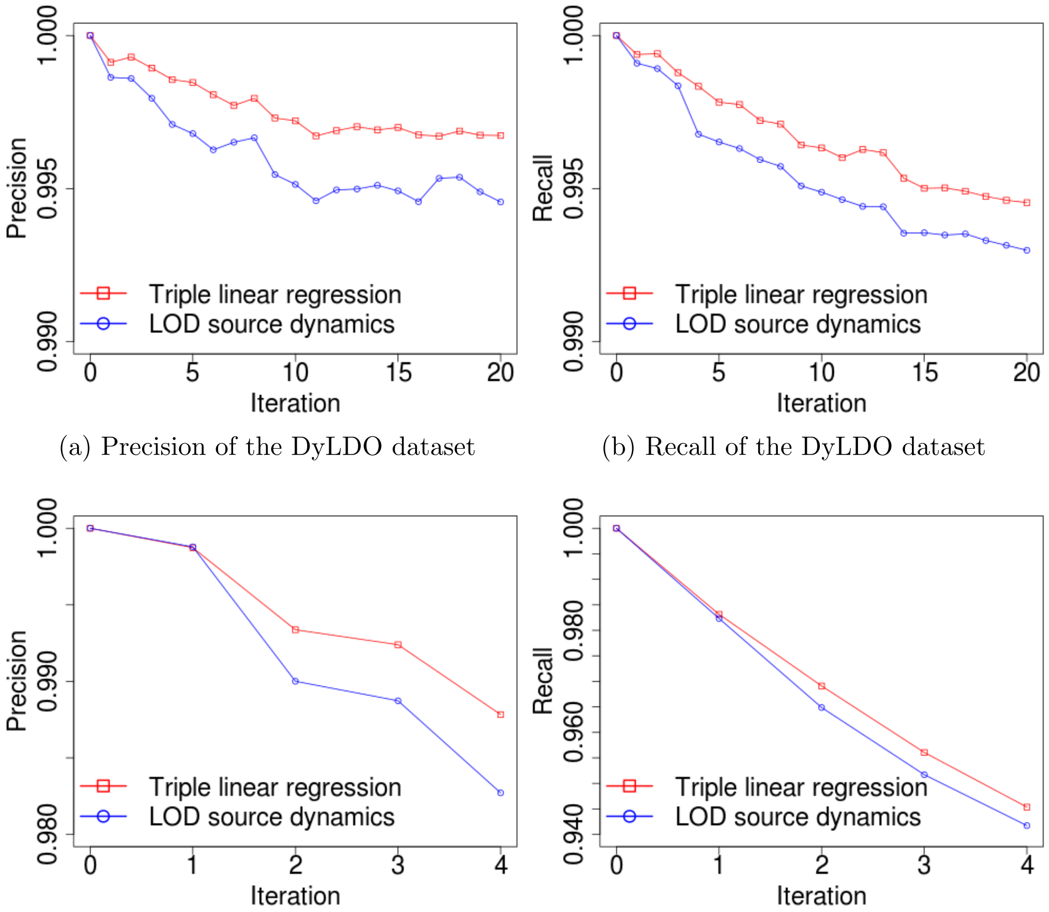
(c) Precision of the Wikidata dataset

(d) Recall of the Wikidata dataset

Figure 7.1: Single-step setup: Precision (left) and recall (right) of the local copies.

disappears with 100% bandwidth, and both strategies achieve a precision of 1.00. Regarding recall, on the other hand, the difference between the two crawling strategies is smaller. However, the novel crawling strategy still performs slightly better.

**Iterative progression** Figure 7.2 shows the results of the iterative progression setup when the relative bandwidth is 20%. The novel crawling strategy always outperforms the other in terms of both precision and recall. In particular, the novel crawling strategy is much better in terms of precision. At the beginning of the iteration, the difference between the two crawling strategies is small and increases. After a few iterations, however, the amount of the difference becomes stable.



(a) Precision of the DyLDO dataset

(b) Recall of the DyLDO dataset

(c) Precision of the Wikidata dataset

(d) Recall of the Wikidata dataset

Figure 7.2: Precision (left) and recall (right) of the resulting local copies in the iterative progression setup with a middle bandwidth (20%).

## 7.5 Discussion

In both setups, we observe that the novel crawling strategy outperforms the baseline. In particular, we note that the novel crawling strategy performes better in the iterative progression setup. The novel crawling strategy has the advantage that once a linear regression model is trained, it does not need any past snapshots to compute preference scores. In contrast, Dividino et al.'s LOD source dynamics [DGS15] requires to update the preference scores using past snapshots after each iteration. In other words, the strategy always needs the latest of the past states of RDF documents to compute how much the RDF documents have been modified. We conjecture that since our novel crawling strategy looks into which triples are included in an RDF document and content of each triple, it captures the dynamics of the RDF documents better. Moreover, we conclude that the linear regression model does generalize since the performance of our strategy does not worsen as it slides over the points in time. However, the model should be updated when many new RDF documents are added or after a long time has passed. Please note that we use a linear regression model due to its simplicity for the crawling strategy. The model is able to capture for how many weeks or months a triple has been alive. In terms of other regression models, we also have tried logistic regression, but it leads to almost the same results. In addition, we experiment with random forest regression [Bre01], boosted tree regression [Fri02], and decision tree regression. The results of the linear regression model are despite its simplicity better than those of the other models.

# Chapter 8

# Application IV: Change Verification for Knowledge Graphs

To keep the information in knowledge graphs up-to-date, many editors contribute to making changes on knowledge graphs such as Wikidata [VK14]. While the majority of changes are correct, knowledge graphs also receive incorrect changes due to vandalism, carelessness, and misunderstanding by editors. Therefore, administrators manually verify these changes [TVS+16]. Thus, the change verification for knowledge graphs is demanding in general. In fact, Tanon et al. [TVS+16] argued that a significant increase in the amount of changes needs to go along with either an increase in the number of administrators or with the provision of tools to improve the present administrators' efficiency. In addition, since even automatically created knowledge graphs such as DBpedia [ABK+07] and YAGO [SKW07] rely on Wikipedia infoboxes made by editors, it is not trivial for them to assess changes.

In this chapter, we develop classifiers for changes to a knowledge graph using the topological features discussed in Section 6.2. Our classifiers compute the scores of changes using those features. A high score indicates that the change is likely to be incorrect and should be rejected. As a dataset, we use the snapshots of Wikidata over two years. The experiment demonstrates that novel topological features are useful to automatically judge whether an incoming change is correct or incorrect. These features are especially useful to classify changes whose objects are a URI. Since in previous studies change verification performed worse for these changes than for the changes that contained literals [HPS+16], our novel features help to complement the existing methods of change verification.

Section 8.1 formalizes the problem of change verification for a knowledge graph. Subsequently, we introduce a method of change verification using novel topological features in Section 8.2. Section 8.3 details the experiment, and Section 8.4 reports the results. Finally, we discuss the results and the efficiency of novel topological features in Section 8.5.

## 8.1 Problem Statement

A change is represented by a tuple, which is composed of a triple $\langle s, p, o \rangle$, a flag $m$, and a time stamp $t$. We compute a score for a change $es(\langle s, p, o \rangle, m, t)$ on a knowledge graph $G$ and classify it as correct or incorrect. A higher score indicates that a change is likely incorrect. A triple $\langle s, p, o \rangle$ consists of subject $s$, predicate $p$, and object $o$. We consider the sets of all URIs $R$ and literals $L$. In a triple $\langle s, p, o \rangle$, a subject $s \in R$ is a URI, a predicate $p \in R$ a URI, and an object $o \in R \cup L$ a URI or a literal. Then, a knowledge graph can be seen as a directed graph, where each node is a subject or object. The set of edges in the graph are considered as triples, which are described as $E = R \times R \times (R \cup L)$. A flag $m$ indicates whether a triple is added ($m = 1$) or deleted ($m = -1$). A time stamp $t$ refers to a point in time at which a change is made.

## 8.2 Change Verifiers

Based on the investigation in Section 6.2, this section develops classifiers that verify whether an incoming change is correct or incorrect. Section 8.2.1 summarizes the features employed by the classifiers. Thereafter, Section 8.2.2 describes the classification algorithms.

### 8.2.1 Features

Table 8.1 summarizes the features used for the classifiers. The first and second columns show the groups and its features, respectively. The topological features are based on Section 6.2. URI out-degree and literal out-degree refer to the number of edges that are connected to a URI node and literal, respectively. In addition, we also employ “predicate” as feature, as done by Tan et al. [TAI+14] and Heindorf et al. [HPS+16]. We convert predicates into features using one-hot encoding. While all 16 features can be used for changes whose objects are a URI, the features from the group “object” cannot be employed for changes whose objects are a literal. Thus, only 10 features are used for these changes.

Table 8.1: Features used by the classifiers for automatic change verification.

| Group | Feature |
|---|---|
| subject | in-degree |
| | out-degree |
| | URI out-degree |
| | literal out-degree |
| | age |
| | last edit |
| predicate | age |
| | last edit |
| | predicate |
| object | in-degree |
| | out-degree |
| | URI out-degree |
| | literal out-degree |
| | age |
| | last edit |
| others | flag $m$ |

### 8.2.2 Classification Algorithms

Tan et al. [TAI+14] observed that logistic regression outperformed GradBoost [DS09] and perceptron [FS99]. Moreover, in their pilot experiments, Heindorf et al. [HPS+16] found that random forest [Bre01] outperformed logistic regression as well as naive Bayes. Therefore, we employ random forest as well as logistic regression in our experiment. We use implementations provided by Turi[1].

**Logistic regression** To avoid overfitting, we use L2 regularization with $\lambda = 0.01$ for all the datasets. We optimize $\lambda$ by 10-fold cross-validation on the training data.

**Random forest** We optimize the maximal tree depth as 8 by 10-fold cross-validation on the training data.

## 8.3 Experiment

We conduct an experiment to investigate the performance of novel topological features for change verification. Section 8.3.1 describes the dataset used in the experiment. Subsequently, Section 8.3.2 introduces the metrics.

[1] `https://turi.com/products/create/docs/graphlab.toolkits.classifier.html`, last accessed on 08/31/2017

Table 8.2: The dataset for training and test. The fourth column provides the number of changes and the fifth shows the rate of positive samples (i .e., incorrect changes) in each dataset.

| Dataset | From | To | # changes | Rate |
|---|---|---|---|---|
| **URI dataset** | | | | |
| training | 04/20/2014 | 01/04/2016 | 90,234,704 | 7.63% |
| test | 01/04/2016 | 06/21/2016 | 22,649,334 | 5.41% |
| **literal dataset** | | | | |
| training | 04/20/2014 | 01/04/2016 | 47,532,819 | 8.01% |
| test | 01/04/2016 | 06/21/2016 | 8,790,632 | 4.67% |

### 8.3.1 Dataset

The experiment uses the Wikidata dataset described in Section 6.2.1. We split the dataset into changes whose objects are a URI and changes whose objects are a literal. We refer to the changes whose objects are a URI as the URI dataset, and to those whose objects are a literal as the literal dataset. We use the split because the different features can be applied to the two datasets. In line with Heindorf et al. [HPS+16], we further divide the two datasets for training and test by time. Table 8.2 provides a description of the datasets. In both datasets, 80% of changes are used for training, and 20% for test.

### 8.3.2 Metrics

To assess how well the classifiers detect incorrect changes, we follow Heindorf et al. [HPS+16] and use two metrics: the area under the curve of the receiver operating characteristic (ROC), and the area under the precision-recall curve (PR). While ROC is used to evaluate classification performance in general, PR provides a different view of imbalanced datasets [DG06]. Please note that we treat incorrect changes as positive and correct ones as negative, in line with Heindorf et al. [HPS+16]. Thus, precision and recall are defined as the fraction of predicted incorrect changes that are truly incorrect, and the fraction of all truly incorrect changes that are identified, respectively.

## 8.4 Results

Table 8.3 provides the results of the classification with respect to the datasets. In addition, Figure 8.1 shows the corresponding PR curves. In Table 8.3 and Figure 8.1, the classifiers perform well for the URI dataset, but not for the literal

Table 8.3: Classification result of the change verification using the test data. Metrics are the area under the curve of the ROC and the PR.

**(a) URI dataset**

| | **ROC** | **PR** |
|---|---|---|
| logistic regression | 0.8350 | 0.3248 |
| random forest | 0.9183 | 0.4728 |

**(b) literal dataset**

| | **ROC** | **PR** |
|---|---|---|
| logistic regression | 0.6543 | 0.0116 |
| random forest | 0.4688 | 0.0043 |

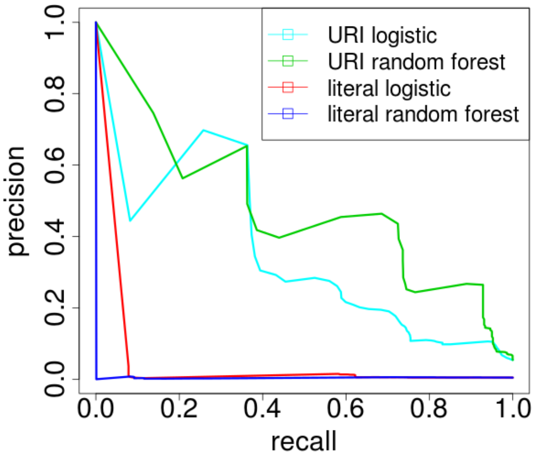


Figure 8.1: PR curves of the classifiers.

dataset. Since the state of the art [HPS+16] does not work well for assessing changes whose objects are a URI, we believe that the novel topological features complement previous works. In terms of the classifiers, while the random forest performs better for the URI dataset, the logistic regression performs better for the literal dataset. A possible reason for the poor performance for the literal dataset is that literals are inherently not counted as nodes in graphs. Thus, they do not follow patterns of the data dynamics such as the preferential attachment discussed in Section 6.2.

To further assess the influence of each feature, we conduct a feature ablation analysis by removing from the classifier one feature at a time as Tan et al. [TAI+14] did. As classification algorithm, we use the random forest for the URI dataset, and the logistic regression for the literal dataset. The third and fourth columns of Table 8.4 show ROC when each feature is not employed. A smaller value indicates that the feature has a larger positive influence. In both datasets, the predicate

Table 8.4: Result of feature ablation analysis. The third and fourth columns show computed ROC when the feature is not used for the classifiers.

| Group | Feature | URI | literal |
|---|---|---|---|
| subject | in-degree | 0.7883 | 0.6193 |
| | out-degree | 0.7915 | 0.6589 |
| | URI out-degree | 0.7884 | 0.6642 |
| | literal out-degree | 0.7946 | 0.6580 |
| | age | 0.7769 | 0.6310 |
| | last edit | 0.9074 | 0.6393 |
| predicate | age | 0.6138 | 0.4500 |
| | last edit | 0.7409 | 0.6163 |
| | predicate | 0.7601 | 0.6233 |
| object | in-degree | 0.7879 | - |
| | out-degree | 0.8929 | - |
| | URI out-degree | 0.8955 | - |
| | literal out-degree | 0.8880 | - |
| | age | 0.8240 | - |
| | last edit | 0.7713 | - |
| others | flag $m$ | 0.7853 | 0.6510 |

age has the largest influence, while the features relevant to out-degree have the smallest influence.

## 8.5 Discussion

To the best of our knowledge, the state of the art in change verification is the work by Heindorf et al. [HPS+16]. They used the WDVC dataset [HPS+15] based on Wikidata for their evaluation, and reported 0.981 of ROC and 0.171 of PR in the Wikidata item body (i .e., a part of a Wikidata article corresponding to triples). We cannot use the WDVC dataset, since the overlap period between the WDVC dataset [HPS+15] and the used snapshots is short. Although a direct comparison with our study is impossible, ROC of our novel change classifier is worse than theirs, while PR of our classifier is better. The difference between the two metrics is that while ROC takes into account true negatives, PR does not. Thus, the novel change classifier judges correct changes as incorrect ones, but detects incorrect changes well. In addition, since we do not use editors' information, the novel change classifier can be applied to new editors as well. In summary, the novel topological features can improve the state of the art.

Regarding the heuristic that labels changes as correct or incorrect (i .e., a change is labeled as incorrect if it is reverted in four weeks), we manually inspect

400 randomly sampled changes in the test data of the URI dataset. The heuristic labels 23 of them as incorrect. From the sampled changes, we find only 1 false positive (falsely labeled as incorrect) and 18 false negatives. Since the rate of false positives is small, we believe that the experiment properly evaluates the performance of detecting incorrect changes.

# Chapter 9

# Conclusion

In this thesis, we confirmed that knowledge graphs can assist in profiling methods, and that profiling methods can capture the data dynamics of knowledge graphs and contribute to their integrity. In Section 9.1, we first reflect on and summarize the insights gained in this thesis. Then, we discuss open issues and possible future areas of study in Section 9.2.

## 9.1 Insights Gained

In the experiment regarding recommending relevant researchers presented in Chapter 4, the profiling methods using knowledge graphs did not work well for the computer science dataset. We conjectured that the reason for this was the quality of the knowledge graph (i .e., the ACM CCS). The ACM CCS contains much fewer entities than the MeSH does. In addition, we used the ACM CCS published in 2012, although the experiment was conducted in 2015. In contrast, the MeSH is updated every week. This time difference might also be a reason for our results. Therefore, it is necessary to check the quality of the knowledge graph before applying it to a profiling method. To this end, Färber et al. [FEM+16] provided different measures to assess the quality of knowledge graphs.

While we investigated profiling the data dynamics of knowledge graphs, we noticed that it was important to sample and preserve a representative knowledge graph. In 2012, Käfer et al. [KUH+12] launched the DyLDO for this reason, and started to collect weekly snapshots of knowledge graphs. The seed list of the DyLDO contains both representative data sources and randomly chosen ones. However, although new knowledge graphs (i .e., data sources) continuously become available and other knowledge graphs go offline, the seed list used by the DyLDO has not been updated or extended since the beginning of 2012. In fact, the size of the weekly snapshots of the DyLDO dataset has decreased by over 50% since it was launched. This produces a bias, because as more time passes, more data sources in the seed list permanently disappear. The Billion Triples Challenge (BTC) datasets provided by the Karlsruhe Institute of Technology provide larger snapshots of knowledge graphs, but their crawling frequency is low (i .e., one snapshot per year). In addition, their seed list is changed with every snapshot, making proper comparisons impossible. Therefore, countermeasures are needed to ensure representativity of the snapshots over time for future researches.

## 9.2 Future Directions

This thesis showed that knowledge graphs are dynamic over time. However, the existing profiling methods using knowledge graphs exploit a static knowledge graph that is captured at a certain point in time. Since documents and social media items have different time stamps, we should use the knowledge graph that corresponds to a given time stamp. For example, a knowledge graph from 2013 should be used to analyze document published in 2013. In addition, in this thesis we assumed that microblog postings mainly describe what is happening and what a user is interested in at a time when a microblog posting is published. However, according to Jatowt et al. [JAK+15], many microblog postings contain temporal expressions that refer to the past or the future. Since these microblog postings reflect user's expectations or memories, we need to examine temporal expressions in the future.

In terms of temporal information in knowledge graphs, we assumed that triples contained in a current knowledge graph are correct and reflect the current state of the world. However, this assumption is not always applicable. For example, we obtain two results, `dbr:Cleveland_Cavaliers` and `dbr:Miami_Heat`, by querying the team of `dbr:LeBron_James` in the DBpedia SPARQL Endpoint. The fact that `dbr:LeBron_James` plays for the `dbr:Cleveland_Cavaliers` is valid for the period between 2003 and 2010, and then again since 2014. On the other hand, his playing for the `dbr:Miami_Heat` is valid between 2010 and 2014. Therefore, the degree of the association between LeBron James and each of two teams varies depending on time. Therefore, it is necessary to annotate temporal information (i .e., time frame in which a triple is valid) to triples that are valid only for a certain period of time. To tackle this problem, YAGO2 [HSB+13] extends a traditional knowledge graph with temporal information as well as spatial information. In addition, Wikidata [VK14] allows temporal information to be stored for each triple. However, a large number of triples in those knowledge graphs are still missing temporal information. Although Talukdar et al. [TWM12] and Jiang et al. [JLG+16] proposed methods to automatically detect triples with temporal information, these methods require manual inputs about constraints regarding that information (e .g., there is only one U.S. president at each point in time). Thus, fully automatic and scalable methods are demanding.

Furthermore, different knowledge graphs have started to annotate certainty (i .e., probability) to each triple. For example, Probase [WLW+12] and Google Knowledge Vault [DGH+14] extract triples from documents and store them with certainty. In addition, Wikidata [VK14] allows conflicting triples to coexist, since many facts in the real world are disputed or simply uncertain. Therefore, in the future, the profiling methods using knowledge graphs need to take into account the uncertainty of each triple when they reveal relevant entities.

# Bibliography

[ABK+07] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. “DBpedia: A nucleus for a web of open data”. In: *International Semantic Web Conference and Asian Semantic Web Conference (ISWC/ASWC).* Springer, 2007, pp. 722–735.

[AH11] L. Averell and A. Heathcote. “The form of the forgetting curve and the fate of memories”. In: *Journal of Mathematical Psychology* 55.1 (2011), pp. 25–35.

[AHK11] F. Abel, E. Herder, and D. Krause. “Extraction of Professional Interests from Social Web Profiles”. In: *Workshop on Augmenting User Models with Real World Experiences to Enhance Personalization and Adaptation (AUM).* 2011, No. 9.

[AS14] C. Aggarwal and K. Subbian. “Evolutionary network analysis: A survey”. In: *ACM Computing Surveys (CSUR)* 47.1 (2014), No. 10.

[AT97] R. B. Anderson and R. D. Tweney. “Artifactual power curves in forgetting”. In: *Memory & Cognition* 25.5 (1997), pp. 724–730.

[AZS+13] M. Acosta, A. Zaveri, E. Simperl, D. Kontokostas, S. Auer, and J. Lehmann. “Crowdsourcing linked data quality assessment”. In: *International Semantic Web Conference (ISWC).* Springer. 2013, pp. 260–276.

[BA99] A.-L. Barabási and R. Albert. “Emergence of Scaling in Random Networks”. In: *Science* 286.5439 (1999), pp. 509–512.

[BC94] D. J. Berndt and J. Clifford. “Using dynamic time warping to find patterns in time series”. In: *International Conference on Knowledge Discovery and Data Mining workshop.* AAAI. 1994, pp. 359–370.

[BDO95] M. W. Berry, S. T. Dumais, and G. W. O’Brien. “Using linear algebra for intelligent information retrieval”. In: *SIAM Review* 37.4 (1995), pp. 573–595.

[BEP+08] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. “Freebase: A collaboratively created graph database for structuring human knowledge”. In: *International Conference on Management of Data (SIGMOD)*. ACM. 2008, pp. 1247–1250.

[BHB09] C. Bizer, T. Heath, and T. Berners-Lee. “Linked Data: The story so far”. In: *Semantic Services, Interoperability and Web Applications: Emerging Concepts*. IGI Global, 2009, pp. 205–227.

[BHK98] J. S. Breese, D. Heckerman, and C. Kadie. “Empirical analysis of predictive algorithms for collaborative filtering”. In: *Conference on Uncertainty in Artificial Intelligence (UAI)*. Morgan Kaufmann, 1998, pp. 43–52.

[BHL01] T. Berners-Lee, J. Hendler, and O. Lassila. “The semantic web”. In: *Scientific American* 284.5 (2001), pp. 28–37.

[BKT+14] G. E. A. P. A. Batista, E. J. Keogh, O. M. Tataw, and V. M. A. de Souza. “CID: An efficient complexity-invariant distance for time series”. In: *Data Mining and Knowledge Discovery* 28.3 (2014), pp. 634–669.

[BL06] D. M. Blei and J. D. Lafferty. “Dynamic topic models”. In: *International Conference on Machine Learning (ICML)*. ACM. 2006, pp. 113–120.

[BNJ03] D. M. Blei, A. Y. Ng, and M. I. Jordan. “Latent Dirichlet allocation”. In: *Journal of Machine Learning Research (JMLR)* 3 (2003), pp. 993–1022.

[BOH12] S. Bostandjiev, J. O’Donovan, and T. Höllerer. “Taste-Weights: A visual interactive hybrid recommender system”. In: *Conference on Recommender Systems (RecSys)*. ACM, 2012, pp. 35–42.

[BP98] S. Brin and L. Page. “The anatomy of a large-scale hypertextual web search engine”. In: *Computer Networks and ISDN Systems* 30.1 (1998), pp. 107–117.

[Bre01] L. Breiman. “Random forests”. In: *Machine Learning* 45.1 (2001), pp. 5–32.

[BWP11] E. Bart, M. Welling, and P. Perona. “Unsupervised organization of image collections: Taxonomies and beyond”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.11 (2011), pp. 2302–2315.

[CBG+09] J. Chang, J. L. Boyd-Graber, S. Gerrish, C. Wang, and D. M. Blei. “Reading tea leaves: How humans interpret topic models.” In: *Conference on Neural Information Processing Systems (NIPS)*. Curran Associates, Inc. 2009, pp. 288–296.

[CG00] J. Cho and H. Garcia-Molina. “The Evolution of the Web and Implications for an Incremental Crawler”. In: *International Conference on Very Large Data Bases (VLDB)*. Morgan Kaufmann Publishers Inc., 2000, pp. 200–209.

[CG03] J. Cho and H. Garcia-Molina. “Effective page refresh policies for web crawlers”. In: *ACM Transactions on Database Systems (TODS)* 28.4 (2003), pp. 390–426.

[CL75] A. M. Collins and E. F. Loftus. “A spreading-activation theory of semantic processing”. In: *Psychological Review* 82.6 (1975), pp. 407–428.

[CNN+10] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi. “Short and tweet: Experiments on recommending content from information streams”. In: *Conference on Human Factors in Computing Systems (CHI)*. ACM. 2010, pp. 1185–1194.

[CS08] R. Crane and D. Sornette. “Robust dynamic classes revealed by measuring the response function of a social system”. In: *Proceedings of the National Academy of Sciences (PNAS)* 105.41 (2008), pp. 15649–15653.

[Das98] V. S. Dasan. *Personalized information retrieval using user-defined profile*. US Patent 5,761,662. 1998.

[DDD+10] T. De Pessemier, S. Dooms, T. Deryckere, and L. Martens. “Time dependency of data quality for collaborative filtering algorithms”. In: *Conference on Recommender Systems (RecSys)*. ACM. 2010, pp. 281–284.

[DDF+90] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. “Indexing by latent semantic analysis”. In: *Journal of the American Society for Information Science* 41.6 (1990), pp. 391–407.

[DG06] J. Davis and M. Goadrich. “The relationship between precision-recall and ROC Curves”. In: *International Conference on Machine Learning (ICML)*. ACM, 2006, pp. 233–240.

[DGH+14] 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: *International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM. 2014, pp. 601–610.

[DGL12] G. De Francisci Morales, A. Gionis, and C. Lucchese. “From chatter to headlines: Harnessing the real-time web for personalized news recommendation”. In: *International Conference on Web Search and Data Mining (WSDM)*. ACM. 2012, pp. 153–162.

[DGS+14] R. Dividino, T. Gottron, A. Scherp, and G. Gröner. “From changes to dynamics: Dynamics analysis of linked open data sources”. In: *International Workshop on Dataset Profiling and Federated Search for Linked Data (PROFILES)*. CEUR, 2014.

[DGS15] R. Dividino, T. Gottron, and A. Scherp. “Strategies for efficiently keeping local Linked Open Data caches up-to-date”. In: *International Semantic Web Conference (ISWC)*. Springer, 2015, pp. 356–373.

[DKG14] R. Dividino, A. Kramer, and T. Gottron. “An investigation of HTTP header information for detecting changes of Linked Open Data sources”. In: *Extended Semantic Web Conference (ESWC) Satellite Events*. Springer, 2014, pp. 199–203.

[DL05] Y. Ding and X. Li. “Time weight collaborative filtering”. In: *International Conference on Information and Knowledge Management (CIKM)*. ACM. 2005, pp. 485–492.

[DS09] J. Duchi and Y. Singer. “Boosting with structural sparsity”. In: *International Conference on Machine Learning (ICML)*. ACM. 2009, pp. 297–304.

[DSG+13] R. Dividino, A. Scherp, G. Gröner, and T. Gottron. “Change-a-LOD: Does the schema on the linked data cloud change or not?” In: *International Conference on Consuming Linked Data (COLD)*. CEUR. 2013.

[DTW+12] Y. Dong, J. Tang, S. Wu, J. Tian, N. V. Chawla, J. Rao, and H. Cao. “Link prediction and recommendation across heterogeneous social networks”. In: *International Conference on Data Mining (ICDM)*. IEEE. 2012, pp. 181–190.

[EA12] P. Esling and C. Agon. “Time-series data mining”. In: *ACM Computing Surveys (CSUR)* 45.1 (2012), No. 12.

[Ebb13] H. Ebbinghaus. *Memory: A contribution to experimental psychology.* Columbia University, 1913.

[Ebb85] H. Ebbinghaus. *Über das Gedächtnis.* Columbia University, 1885.

[EGK+14] F. Erxleben, M. Günther, M. Krötzsch, J. Mendez, and D. Vrandečić. “Introducing Wikidata to the linked data web”. In: *International Semantic Web Conference (ISWC).* Springer. 2014, pp. 50–65.

[EV03] M. Eirinaki and M. Vazirgiannis. “Web mining for web personalization”. In: *ACM Transactions on Internet Technology (TOIT)* 3.1 (2003), pp. 1–27.

[FDK16] M. Francis-Landau, G. Durrett, and D. Klein. “Capturing semantic similarity for entity linking with convolutional neural networks”. In: *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL).* ACL, 2016, pp. 1256–1261.

[FEB+02] A. M. Ferman, J. H. Errico, P. van Beek, and M I. Sezan. “Content-based filtering and personalization using structured metadata”. In: *Joint Conference on Digital Libraries (JCDL).* ACM. 2002, p. 393.

[Fel98] C. Fellbaum. *WordNet.* Wiley Online Library, 1998.

[FEM+16] M. Färber, B. Ell, C. Menne, A. Rettinger, and F. Bartscherer. “Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO”. In: *Semantic Web* Preprint.Preprint (2016), pp. 1–53.

[Fri02] J. H. Friedman. “Stochastic gradient boosting”. In: *Computational Statistics & Data Analysis* 38.4 (2002), pp. 367–378.

[FRM94] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. “Fast subsequence matching in time-series databases”. In: *International Conference on Management of Data (SIGMOD).* ACM, 1994, pp. 419–429.

[FS99] Y. Freund and R. E. Schapire. “Large margin classification using the perceptron algorithm”. In: *Machine Learning* 37.3 (1999), pp. 277–296.

[FW14] W. Feng and J. Wang. “We can learn your #hashtags: Connecting tweets to explicit topics”. In: *International Conference on Data Engineering (ICDE).* IEEE. 2014, pp. 856–867.

[FWR+15] M. Farajtabar, Y. Wang, M. G. Rodriguez, S. Li, H. Zha, and L. Song. “COEVOLVE: A joint point process model for information diffusion and network co-evolution”. In: *Conference on Neural Information Processing Systems (NIPS).* Curran Associates, Inc. 2015, pp. 1954–1962.

[GG14] T. Gottron and C. Gottron. “Perplexity of index models over evolving linked data”. In: *Extended Semantic Web Conference (ESWC).* Springer. 2014, pp. 161–175.

[GGL+16] O.-E. Ganea, M. Ganea, A. Lucchi, C. Eickhoff, and T. Hofmann. “Probabilistic bag-of-hyperlinks model for entity linking”. In: *International Conference on World Wide Web (WWW).* IW3C2. 2016, pp. 927–938.

[GGS+12] C. Guéret, P. Groth, C. Stadler, and J. Lehmann. “Assessing linked data mappings using network measures”. In: *European Semantic Web Conference (ESWC).* Springer. 2012, pp. 87–102.

[GGW03] P. Ganesan, H. Garcia-Molina, and J. Widom. “Exploiting hierarchical domain structure to compute similarity”. In: *ACM Transactions on Information Systems (TOIS)* 21.1 (2003), pp. 64–93.

[GIF+11] F. Goossen, W. IJntema, F. Frasincar, F. Hogenboom, and U. Kaymak. “News personalization using the CF-IDF semantic recommender”. In: *International Conference on Web Intelligence, Mining and Semantics (WIMS).* ACM. 2011, No. 10.

[GNS15a] G. Grosse-Bölting, C. Nishioka, and A. Scherp. “Generic process for extracting user profiles from social media using hierarchical knowledge bases”. In: *International Conference on Semantic Computing (ICSC).* IEEE, 2015, pp. 197–200.

[GNS15b] G. Große-Bölting, C. Nishioka, and A. Scherp. “A comparison of different strategies for automated semantic document annotation”. In: *International Conference on Knowledge Capture (K-CAP).* ACM. 2015, No. 8.

[GO08] C. Grimes and S. O’Brien. “Microscale evolution of web pages”. In: *International conference on World Wide Web (WWW).* ACM. 2008, pp. 1149–1150.

[GS04] T. L. Griffiths and M. Steyvers. “Finding scientific topics”. In: *National Academy of Sciences (NAS)* 101.suppl 1 (2004), pp. 5228–5235.

[Haz10] T. J. Hazen. “Direct and latent modeling techniques for computing spoken document similarity”. In: *Spoken Language Technology Workshop (SLT)*. IEEE. 2010, pp. 366–371.

[HB11] T. Heath and C. Bizer. *Linked data: Evolving the web into a global data space*. Morgan & Claypool Publishers, 2011.

[HD10] L. Hong and B. D. Davison. “Empirical study of topic modeling in Twitter”. In: *Workshop on Social Media Analytics (SOMA)*. ACM. 2010, pp. 80–88.

[Hof99] T. Hofmann. “Probabilistic latent semantic indexing”. In: *International Conference on Research and Development in Information Retrieval (SIGIR)*. ACM. 1999, pp. 50–57.

[HPS+15] S. Heindorf, M. Potthast, B. Stein, and G. Engels. “Towards vandalism detection in knowledge bases: Corpus construction and analysis”. In: *International Conference on Research and Development in Information Retrieval (SIGIR)*. ACM. 2015, pp. 831–834.

[HPS+16] S. Heindorf, M. Potthast, B. Stein, and G. Engels. “Vandalism detection in Wikidata”. In: *International Conference on Information and Knowledge Management (CIKM)*. ACM. 2016, pp. 327–336.

[HSB+13] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. “YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia”. In: *Artificial Intelligence* 194 (2013), pp. 28–61.

[JAK+15] A. Jatowt, É. Antoine, Y. Kawai, and T. Akiyama. “Mapping temporal horizons: Analysis of collective future and past related attention in Twitter”. In: *International Conference on World Wide Web (WWW)*. ACM. 2015, pp. 484–494.

[JLG+16] T. Jiang, T. Liu, T. Ge, L. Sha, B. Chang, S. Li, and Z. Sui. “Towards time-aware knowledge graph completion”. In: *International Conference on Computational Linguistics (COLING)*. ACL. 2016, pp. 1715–1724.

[JWR00] K. S. Jones, S. Walker, and S. E. Robertson. “A probabilistic model of information retrieval: Development and comparative experiments: Part 2”. In: *Information processing & management* 36.6 (2000), pp. 809–840.

[JZF+10] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich. *Recommender systems: An introduction*. Cambridge University Press, 2010.

[KAU+13] T. Käfer, A. Abdelrahman, J. Umbrich, P. O’Byrne, and A. Hogan. “Observing linked data dynamics”. In: *Extended Semantic Web Conference (ESWC)*. Springer, 2013, pp. 213–227.

[KB06] S. Y. X. Komiak and I. Benbasat. “The effects of personalization and familiarity on trust and adoption of recommendation agents”. In: *MIS Quarterly* 30.4 (2006), pp. 941–960.

[Keo02] E. Keogh. “Exact indexing of dynamic time warping”. In: *International Conference on Very Large Data Bases (VLDB)*. VLDB Endowment. 2002, pp. 406–417.

[KH10] A. M. Kaplan and M. Haenlein. “Users of the world, unite! The challenges and opportunities of social media”. In: *Business Horizons* 53.1 (2010), pp. 59–68.

[KJN08] M. K. Khribi, M. Jemni, and O. Nasraoui. “Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval”. In: *International Conference on Advanced Learning Technologies (ICALT)*. IEEE. 2008, pp. 241–245.

[KJV+14] P. Kapanipathi, P. Jain, C. Venkataramani, and A. Sheth. “User interests identification on Twitter using a hierarchical knowledge base”. In: *Extended Semantic Web Conference (ESWC)*. Springer, 2014, pp. 99–113.

[KUH+12] T. Käfer, J. Umbrich, A. Hogan, and A. Polleres. “Towards a dynamic linked data observatory”. In: *Workshop on Linked Data on the Web (LDOW)*. CEUR, 2012.

[LBK+08] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. “Microscopic evolution of social networks”. In: *International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM. 2008, pp. 462–470.

[LKF05] J. Leskovec, J. Kleinberg, and C. Faloutsos. "Graphs over time: Densification laws, shrinking diameters and possible explanations". In: *International Conference on Knowledge Discovery and Data Mining (KDD).* ACM. 2005, pp. 177–187.

[LLB+09] M. Luther, T. Liebig, S. Böhm, and O. Noppens. "Who the heck is the father of Bob?" In: *European Semantic Web Conference (ESWC).* Springer. 2009, pp. 66–80.

[LLZ12] C. Lu, W. Lam, and Y. Zhang. "Twitter user modeling and tweets recommendation based on Wikipedia concept graph". In: *AAAI Workshop on Intelligent Techniques for Web Personalization and Recommendation.* AAAI. 2012, pp. 33–38.

[LMC11] N. Lao, T. Mitchell, and W. W. Cohen. "Random walk inference and learning in a large scale knowledge base". In: *Conference on Empirical Methods in Natural Language Processing (EMNLP).* ACL. 2011, pp. 529–539.

[LN89] D. C. Liu and J. Nocedal. "On the limited memory BFGS method for large scale optimization". In: *Mathematical Programming* 45.1 (1989), pp. 503–528.

[LPB+10] J. Letierce, A. Passant, J. G. Breslin, and S. Decker. "Understanding how Twitter is used to spread scientific messages". In: *International Web Science Conference (WebSci).* Web Science Trust. 2010.

[MDL+00] B. Mobasher, H. Dai, T. Luo, Y. Sun, and J. Zhu. "Integrating web usage and content mining for more effective personalization". In: *International Conference on Electronic Commerce and Web Technologies.* Springer, 2000, pp. 165–176.

[MDS01] S. E. Middleton, D. C. De Roure, and N. R. Shadbolt. "Capturing knowledge of user preferences: Ontologies in recommender systems". In: *International Conference on Knowledge Capture (K-CAP).* ACM. 2001, pp. 100–107.

[Men80] J. L. Mendoza. "A significance test for multisample sphericity". In: *Psychometrika* 45.4 (1980), pp. 495–498.

[MJG+11] P. N. Mendes, M. Jakob, A. Garcia-Silva, and C. Bizer. "DBpedia Spotlight: Shedding light on the web of documents". In: *International Conference on Semantic Systems (I-Semantics).* ACM. 2011, pp. 1–8.

[Mob07] B. Mobasher. “Data mining for web personalization”. In: *The Adaptive Web*. Springer, 2007, pp. 90–135.

[MRS08] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Vol. 1. Cambridge University Press, 2008.

[MUA10] M. Martin, J. Unbehauen, and S. Auer. “Improving the performance of semantic web applications with SPARQL query caching”. In: *Extended Semantic Web Conference (ESWC)*. Springer. 2010, pp. 304–318.

[New01] M. E. J. Newman. “Clustering and preferential attachment in growing networks”. In: *Physical Review E* 64.2 (2001), No. 025102.

[NGS15] C. Nishioka, G. Große-Bölting, and A. Scherp. “Influence of time on user profiling and recommending researchers in social media”. In: *International Conference on Knowledge Technologies and Data-driven Business (i-KNOW)*. ACM. 2015, No. 9.

[NS15] C. Nishioka and A. Scherp. “Temporal patterns and periodicity of entity dynamics in the Linked Open Data cloud”. In: *International Conference on Knowledge Capture (K-CAP)*. ACM. 2015, No. 22.

[NS16] C. Nishioka and A. Scherp. “Profiling vs. time vs. content: What does matter for top-k publication recommendation based on Twitter profiles?” In: *Joint Conference on Digital Libraries (JCDL)*. ACM. 2016, pp. 171–180.

[NS17] C. Nishioka and A. Scherp. “Keeping linked open data caches up-to-date by predicting the life-time of RDF triples”. In: *International Conference on Web Intelligence (WI)*. ACM. 2017, pp. 73–80.

[NS18] C. Nishioka and A. Scherp. “Analysing the evolution of knowledge graphs for the purpose of change verification”. In: *IEEE International Conference on Semantic Computing (ICSC)*. IEEE. 2018.

[NSD15] C. Nishioka, A. Scherp, and K. Dellschaft. “Comparing tweet classifications by authors’ hashtags, machine learning, and human annotators”. In: *International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*. IEEE. 2015, pp. 67–74.

[NTK12] M. Nickel, V. Tresp, and H.-P. Kriegel. “Factorizing YAGO: Scalable machine learning for linked data”. In: *International Conference on World Wide Web (WWW)*. ACM. 2012, pp. 271–280.

[NXC+16] Y. Ni, Q. K. Xu, F. Cao, Y. Mass, D. Sheinwald, H. J. Zhu, and S. S. Cao. "Semantic documents relatedness using concept graph representation". In: *International Conference on Web Search and Data Mining (WSDM)*. ACM, 2016, pp. 635–644.

[OBP12] F. Orlandi, J. Breslin, and A. Passant. "Aggregated, interoperable and multi-domain user profiles for the social web". In: *International Conference on Semantic Systems (I-SEMANTICS)*. ACM. 2012, pp. 41–48.

[Orl14] F. Orlandi. "Profiling user interests on the social semantic web". PhD thesis. Digital Enterprise Research Institute (DERI), National University of Ireland, 2014.

[Pau16] H. Paulheim. "Knowledge graph refinement: A survey of approaches and evaluation methods". In: *Semantic Web* 8.3 (2016), pp. 489–508.

[PB14] H. Paulheim and C. Bizer. "Improving the quality of linked data using statistical distributions". In: *International Journal on Semantic Web and Information Systems (IJSWIS)* 10.2 (2014), pp. 63–86.

[PBM+99] L. Page, S. Brin, R. Motwani, and T. Winograd. *The PageRank citation ranking: Bringing order to the web.* Tech. rep. Stanford InfoLab, 1999.

[PC12] C. Pesquita and F. M. Couto. "Predicting the extension of biomedical ontologies". In: *PLOS Computational Biology* 8.9 (2012), pp. 1–16.

[PG11] M. Pennacchiotti and S. Gurumurthy. "Investigating topic models for social media user recommendation". In: *International Conference on World Wide Web (WWW)*. ACM. 2011, pp. 101–102.

[PG15] J. Paparrizos and L. Gravano. "k-Shape: Efficient and accurate clustering of time series". In: *International Conference on Management of Data (SIGMOD)*. ACM. 2015, pp. 1855–1870.

[PH11] N. Popitsch and B. Haslhofer. "DSNotify–A solution for event detection and link maintenance in dynamic datasets". In: *Web Semantics* 9.3 (2011), pp. 266–283.

[PMA+16] S. Perera, P. N. Mendes, A. Alex, A. P. Sheth, and K. Thirunarayan. "Implicit entity linking in tweets". In: *International Semantic Web Conference (ISWC)*. Springer. 2016, pp. 118–132.

[PMB+11] O. Phelan, K. McCarthy, M. Bennett, and B. Smyth. “Terms of a feather: Content-based news recommendation and discovery using Twitter”. In: *European Conference on Information Retrieval (ECIR)*. Springer. 2011, pp. 448–459.

[PSD00] J. K. Pritchard, M. Stephens, and P. Donnelly. “Inference of population structure using multilocus genotype data”. In: *Genetics* 155.2 (2000), pp. 945–959.

[RB13] K. Radinsky and P. N. Bennett. “Predicting content change on the web”. In: *International Conference on Web Search and Data Mining (WSDM)*. ACM. 2013, pp. 415–424.

[RBN14] J. Rybak, K. Balog, and K. Nørvåg. “Temporal expertise profiling”. In: *European Conference on Information Retrieval (ECIR)*. Springer, 2014, pp. 540–546.

[RCM+12] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh. “Searching and mining trillions of time series subsequences under dynamic time warping”. In: *International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM. 2012, pp. 262–270.

[RHN+09] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. “Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora”. In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL. 2009, pp. 248–256.

[RSG+15] I. S. Ribeiro, R. L. T. Santos, M. A. Gonçalves, and A. H. F. Laender. “On tag recommendation for expertise profiling: A case study in the scientific domain”. In: *International Conference on Web Search and Data Mining (WSDM)*. ACM, 2015, pp. 189–198.

[RT11] G. Rizzo and R. Troncy. “NERD: Evaluating named entity recognition tools in the web of data”. In: *Workshop on Web Scale Knowledge Extraction*. 2011.

[RV13] F. Rousseau and M. Vazirgiannis. “Graph-of-word and TW-IDF: New approach to ad hoc IR”. In: *International Conference on Information and Knowledge Management (CIKM)*. ACM. 2013, pp. 59–68.

[RW94] S. E. Robertson and S. Walker. "Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval". In: *International Conference on Research and Development in Information Retrieval (SIGIR)*. Springer. 1994, pp. 232–241.

[RWB99] S. E. Robertson, S. Walker, and M. Beaulieu. "Okapi at TREC-7: Automatic ad hoc, filtering, VLC and interactive track". In: *TREC-7*. NIST, 1999, pp. 253–264.

[RWJ+94] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. "Okapi at TREC–3". In: *TREC–3*. NIST, 1994, pp. 109–126.

[SB88] G. Salton and C. Buckley. "Term-weighting approaches in automatic text retrieval". In: *Information processing & management* 24.5 (1988), pp. 513–523.

[SBP14] M. Schmachtenberg, C. Bizer, and H. Paulheim. "Adoption of the linked data best practices in different topical domains". In: *International Semantic Web Conference (ISWC)*. Springer, 2014, pp. 245–260.

[SC98] S. J. Soltysiak and I. B. Crabtree. "Automatic learning of user profiles — towards the personalisation of agent services". In: *BT Technology Journal* 16.3 (1998), pp. 110–117.

[SCA+15] A. S. R. Santos, C. R de Carvalho, J. M. Almeida, E. S. de Moura, A. S. da Silva, and N. Ziviani. "A genetic programming framework to schedule webpage updates". In: *Information Retrieval Journal* 18.1 (2015), pp. 73–94.

[SCJ12] P. Sarkar, D. Chakrabarti, and M. Jordan. "Nonparametric link prediction in dynamic networks". In: *International Conference on Machine Learning (ICML)*. Omnipress. 2012, pp. 1687–1694.

[SCM+13] R. Socher, D. Chen, C. D. Manning, and A. Ng. "Reasoning with neural tensor networks for knowledge base completion". In: *Conference on Neural Information Processing Systems (NIPS)*. Curran Associates, Inc. 2013, pp. 926–934.

[Sha86] J. P. Shaffer. "Modified sequentially rejective multiple test procedures". In: *Journal of the American Statistical Association* 81.395 (1986), pp. 826–831.

[SHN15] M. Shirakawa, T. Hara, and S. Nishio. “N-gram IDF: A Global Term Weighting Scheme Based on Information Distance”. In: *International Conference on World Wide Web (WWW).* IW3C2. 2015, pp. 960–970.

[SK10] K. Sugiyama and M.-Y. Kan. “Scholarly paper recommendation via user’s recent research interests”. In: *Joint Conference on Digital Libraries (JCDL).* ACM. 2010, pp. 29–38.

[SKW07] F. M. Suchanek, G. Kasneci, and G. Weikum. “YAGO: A core of semantic knowledge”. In: *International Conference on World Wide Web (WWW).* ACM. 2007, pp. 697–706.

[SM13] S. L. Sangam and S. S. Mogali. “Obsolescence of literature in the field of social sciences”. In: *PEARL* 7.3 (2013), pp. 162–168.

[SM86] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* New York, NY, USA: McGraw-Hill, Inc., 1986.

[SP14a] M. Schuhmacher and S. P. Ponzetto. “Knowledge-based graph document modeling”. In: *International Conference on Web Search and Data Mining (WSDM).* ACM. 2014, pp. 543–552.

[SP14b] M. Schuhmacher and S. P. Ponzetto. “Knowledge-based graph document modeling”. In: *International Conference on Web Search and Data Mining (WSDM).* ACM. 2014, pp. 543–552.

[SRZ+08] J. Sivic, B. C. Russell, A. Zisserman, W. T. Freeman, and A. A. Efros. “Unsupervised discovery of visual object class hierarchies”. In: *Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE. 2008, pp. 1–8.

[SWL+13] W. Shen, J. Wang, P. Luo, and M. Wang. “Linking named entities in tweets with knowledge base via user interest modeling”. In: *International Conference on Knowledge Discovery and Data Mining (KDD).* ACM. 2013, pp. 68–76.

[SWY75] G. Salton, A. Wong, and C. Yang. “A vector space model for automatic indexing”. In: *Communications of the ACM (CACM)* 18.11 (1975), pp. 613–620.

[SZA+13] A. S. R. Santos, N. Ziviani, J. Almeida, C. R. Carvalho, E. S. de Moura, and A. S. da Silva. “Learning to schedule webpage updates using genetic programming”. In: *International Symposium on String Processing and Information Retrieval (SPIRE).* Springer. 2013, pp. 271–278.

[TAI+14] C. H. Tan, E. Agichtein, P. Ipeirotis, and E. Gabrilovich. “Trust, but verify: Predicting contribution quality for knowledge base construction and curation”. In: *International Conference on Web Search and Data Mining (WSDM).* ACM. 2014, pp. 553–562.

[TCL+16] S. Trani, D. Ceccarelli, C. Lucchese, S. Orlando, and R. Perego. “SEL: A unified algorithm for entity linking and saliency detection”. In: *Symposium on Document Engineering (DocEng).* ACM, 2016, pp. 85–94.

[TDH05] J. Teevan, S. T. Dumais, and E. Horvitz. “Personalizing search via automated analysis of interests and activities”. In: *International Conference on Research and Development in Information Retrieval (SIGIR).* ACM. 2005, pp. 449–456.

[TDO07] G. Tummarello, R. Delbru, and E. Oren. “Sindice.com: Weaving the open linked data”. In: *International Semantic Web Conference and Asian Semantic Web Conference (ISWC/ASWC).* Springer, 2007, pp. 552–565.

[TM10] Q. Tan and P. Mitra. “Clustering-based incremental web crawling”. In: *ACM Transactions on Information Systems (TOIS)* 28.4 (2010), No. 17.

[TMH16] Y.-K. Tang, X.-L. Mao, and H. Huang. “Labeled phrase latent Dirichlet allocation”. In: *International Conference on Web Information Systems Engineering (WISE).* Springer. 2016, pp. 525–536.

[Tof81] A. Toffler. *The third wave.* Bantam Books, 1981.

[TVS+16] T. P. Tanon, D Vrandečić, S Schaffert, T Steiner, and L. Pintscher. “From Freebase to Wikidata: The great migration”. In: *International Conference on World Wide Web (WWW).* IW3C2. 2016, pp. 1419–1428.

[TWM12] P. P. Talukdar, D. Wijaya, and T. Mitchell. “Coupled temporal scoping of relational facts”. In: *International Conference on Web Search and Data Mining (WSDM).* ACM. 2012, pp. 73–82.

[UHH+10] J. Umbrich, M. Hausenblas, A. Hogan, A. Polleres, and S. Decker. "Towards dataset dynamics: Change frequency of linked open data sources". In: *Workshop on Linked Data on the Web (LDOW).* CEUR, 2010.

[UKL10] J. Umbrich, M. Karnstedt, and S. Land. "Towards understanding the changing web: Mining the dynamics of linked-data sources and entities". In: *Workshop on Knowledge Discovery, Data Mining, Maschinelles Lernen (KDML).* 2010, pp. 159–162.

[VAP+17] P.-Y. Vandenbussche, G. A Atemezing, M. Poveda-Villalón, and B. Vatant. "Linked Open Vocabularies (LOV): A gateway to reusable semantic vocabularies on the Web". In: *Semantic Web* 8.3 (2017), pp. 437–452.

[VK14] D. Vrandečić and M. Krötzsch. "Wikidata: A free collaborative knowledge base". In: *Communications of the ACM (CACM)* 57.10 (2014), pp. 78–85.

[VKG11] P. Venetis, G. Koutrika, and H. Garcia-Molina. "On the selection of tags for tag clouds". In: *International Conference on Web Search and Data Mining (WSDM).* ACM. 2011, pp. 835–844.

[WB11] C. Wang and D. M. Blei. "Collaborative topic modeling for recommending scientific articles". In: *International Conference on Knowledge Discovery and Data Mining (KDD).* ACM, 2011, pp. 448–456.

[WE91] J. T. Wixted and E. B. Ebbesen. "On the form of forgetting". In: *Psychological Science* 2.6 (1991), pp. 409–415.

[WE97] J. T. Wixted and E. B. Ebbesen. "Genuine power curves in forgetting: A quantitative analysis of individual subject forgetting functions". In: *Memory & cognition* 25.5 (1997), pp. 731–739.

[WLW+12] W. Wu, H. Li, H. Wang, and K. Q. Zhu. "Probase: A probabilistic taxonomy for text understanding". In: *International Conference on Management of Data (SIGMOD).* ACM. 2012, pp. 481–492.

[WMD+13] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh. "Experimental comparison of representation methods and distance measures for time series data". In: *Data Mining and Knowledge Discovery* 26.2 (2013), pp. 275–309.

[XBF+08] S. Xu, S. Bao, B. Fei, Z. Su, and Y. Yu. "Exploring folksonomy for personalized search". In: *International Conference on Research and Development in Information Retrieval (SIGIR).* ACM. 2008, pp. 155–162.

[XGH+12] Y. Xu, X. Guo, J. Hao, J. Ma, R. Y. K. Lau, and W. Xu. "Combining social network and semantic concept analysis for personalized academic researcher recommendation". In: *Decision Support Systems* 54.1 (2012), pp. 564–573.

[YKS+14] S.-H. Yang, A. Kolcz, A. Schlaikjer, and P. Gupta. "Large-scale high-precision topic modeling on Twitter". In: *International Conference on Knowledge Discovery and Data Mining (KDD).* ACM. 2014, pp. 1907–1916.

[YL11] J. Yang and J. Leskovec. "Patterns of temporal variation in online media". In: *International Conference on Web Search and Data Mining (WSDM).* ACM. 2011, pp. 177–186.

[YLH+03] S. S. Yau, H. Liu, D. Huang, and Y. Yao. "Situation-aware personalized information retrieval for mobile internet". In: *International Computer Software and Applications Conference (COMPSAC).* IEEE. 2003, pp. 639–644.

[ZI13] Q. Zheng and H. H. S. Ip. "Effectiveness of the data generated on different time in latent factor model". In: *Conference on Recommender Systems (RecSys).* ACM. 2013, pp. 327–330.

[ZJW+11] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. "Comparing Twitter and traditional media using topic models". In: *European Conference on Information Retrieval (ECIR).* Springer. 2011, pp. 338–349.

[ZL15] Reza Zafarani and Huan Liu. "Evaluation without ground truth in social media research". In: *Communications of the ACM (CACM)* 58.6 (2015), pp. 54–60.

[ZST+15] W. E. Zhang, Q. Z. Sheng, K. Taylor, and Y. Qin. "Identifying and caching hot triples for efficient RDF query processing". In: *International Conference on Database Systems for Advanced Applications (DASFAA).* Springer. 2015, pp. 259–274.

# Appendix A

# Detailed Results of the Recommender System for Scientific Publications

This chapter describes the detailed results of the recommendation performance provided in Section 5.4.

## A.1 Mean Average Precision

Section 5.4 shows the detailed analysis of the results using only rankscore. In contrast, this section provides the detailed analysis using Mean Average Precision (MAP). Average Precision (AP) is calculated as below:

$$AP = \frac{1}{|hits|} \sum_{d \in hits} Precision@rank(d), \tag{A.1}$$

where $hits$ and $rank(d)$ stand for the set of relevant publications and the rank of the publication $d$, respectively. $|hits|$ is the number of relevant publications in the recommendation list. $Precision@rank(d)$ denotes the precision at cut off $rank(d)$ in the recommendation list. Mean Average Precision (MAP) is the mean average of the Average Precision of all subjects. This section first compares the twelve different recommendation strategies. Subsequently, we investigate the influence of the different experimental factors.

Table A.1: Mean Average Precision (MAP) of the strategies in decreasing order. M and SD denote mean and standard deviation, respectively.

| | **Strategy** | | | **MAP** |
|---|---|---|---|---|
| | **Profiling Method** | **Decay Function** | **Content** | **M (SD)** |
| 1. | CF-IDF | Sliding Window | All | .71 (.32) |
| 2. | HCF-IDF | Exponential | All | .65 (.33) |
| 3. | HCF-IDF | Exponential | Title | .65 (.32) |
| 4. | CF-IDF | Exponential | All | .65 (.35) |
| 5. | HCF-IDF | Sliding Window | Title | .65 (.34) |
| 6. | HCF-IDF | Sliding Window | All | .65 (.34) |
| 7. | CF-IDF | Exponential | Title | .58 (.35) |
| 8. | CF-IDF | Sliding Window | Title | .55 (.34) |
| 9. | LDA | Exponential | All | .47 (.39) |
| 10. | LDA | Exponential | Title | .44 (.34) |
| 11. | LDA | Sliding Window | Title | .43 (.35) |
| 12. | LDA | Sliding Window | All | .40 (.42) |

### A.1.1 Best Performing Strategy

Table A.1 shows the Mean Average Precisions (MAP) of the twelve strategies. The order of the strategies is almost same with rankscores shown in Table 5.2. In order to investigate significant differences among strategies, we first apply Mauchly's test and found a violation of sphericity in the strategies ($\chi^2(65) = 353.51$, $p = .00$). Subsequently, we run a one-way repeated-measure ANOVA with a Greenhouse-Geisser correction of $\epsilon = .65$. It reveals a significant difference of the strategies ($F(7.17, 875.15) = 15.59$, $p = .00$). To assess the statistical significance of pairwise differences between the twelve strategies, a post-hoc analysis is performed using Shaffer's MSRB procedure [Sha86]. The result of the post-hoc analysis is presented in Table A.2. The vertical and horizontal dimensions of the table show the eleven-by-eleven comparison of the twelve strategies. As one can see, we observe various significant differences between strategies (marked in bold font).

### A.1.2 Influence of the Three Experimental Factors

Subsequently, we analyze the results with respect to each factor with MAP. First, we apply Mendoza's test [Men80] which shows violations of sphericity against the factors *Profiling Method* × *Temporal Decay Function* ($\chi^2(2) = 10.30$, $p = .01$), and *Profiling Method* × *Publication Content* ($\chi^2(2) = 13.18$, $p = .00$). Thus, we run three-way repeated-measure ANOVA with a Greenhouse-Geisser correction of $\epsilon = .92$ for the factor *Profiling Method* × *Temporal Decay Function* and $\epsilon = .91$ for

Table A.2: Post-hoc analysis of Mean Average Precision (MAP) with p-values for pairwise comparison over the strategies using Shaffer's MSRB procedure. The p-values are marked in bold font if $p < .05$, which indicates a significant difference between the two strategies. Strategies are sorted by MAP as shown in Table A.1.

| | | | | 2. | 3. | 4. | 5. | 6. | 7. | 8. | 9. | 10. | 11. | 12. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | HCF-IDF | HCF-IDF | CF-IDF | HCF-IDF | HCF-IDF | CF-IDF | CF-IDF | LDA | LDA | LDA | LDA |
| | | | | Exponential | Exponential | Exponential | Sliding Window | Sliding Window | Exponential | Sliding Window | Exponential | Exponential | Sliding Window | Sliding Window |
| | | | | All | Title | All | Title | All | Title | Title | All | Title | Title | All |
| 1. | CF-IDF | Sliding Window | All | .99 | .99 | .99 | .99 | .99 | **.02** | **.00** | **.00** | **.00** | **.00** | **.00** |
| 2. | HCF-IDF | Exponential | All | | .99 | .99 | .99 | .99 | .99 | .25 | **.00** | **.00** | **.00** | **.00** |
| 3. | HCF-IDF | Exponential | Title | | | .99 | .99 | .99 | .99 | .99 | **.00** | **.00** | **.00** | **.00** |
| 4. | CF-IDF | Exponential | All | | | | .99 | .99 | .99 | .33 | **.01** | **.00** | **.00** | **.00** |
| 5. | HCF-IDF | Sliding Window | Title | | | | | .99 | .99 | .54 | **.00** | **.00** | **.00** | **.00** |
| 6. | HCF-IDF | Sliding Window | All | | | | | | .99 | .60 | **.00** | **.00** | **.00** | **.00** |
| 7. | CF-IDF | Exponential | Title | | | | | | | .99 | .67 | **.02** | **.01** | **.00** |
| 8. | CF-IDF | Sliding Window | Title | | | | | | | | .99 | .06 | **.03** | **.02** |
| 9. | LDA | Exponential | All | | | | | | | | | .99 | .99 | .64 |
| 10. | LDA | Exponential | Title | | | | | | | | | | .99 | .99 |
| 11. | LDA | Sliding Window | Title | | | | | | | | | | | .99 |

Table A.3: Three-way repeated-measure ANOVA with Greenhouse-Geisser correction with F-ratio, $\eta^2$, and p-value for MAP.

| **Factor** | **F** | $\eta^2$ | **p** |
|---|---|---|---|
| *Profiling Method* | 51.79 | .42 | **.00** |
| *Temporal Decay Function* | 0.33 | .00 | .57 |
| *Publication Content* | 5.16 | .04 | **.02** |
| *Profiling Method* × *Temporal Decay Function* | 1.66 | .01 | .20 |
| *Profiling Method* × *Publication Content* | 4.76 | .02 | **.01** |
| *Temporal Decay Function* × *Publication Content* | 0.02 | .00 | .90 |
| *Profiling Method* × *Temporal Decay Function* × *Publication Content* | 3.19 | .03 | **.04** |

the factor *Profiling Method* × *Publication Content*. Table A.3 shows the results of applying an ANOVA. $\eta^2$ indicates the effect size of each factor. For all the factors that make a significant difference, we conduct a post-hoc analysis using Shaffer's MSRB Procedure.

Subsequently, the post-hoc analyses with respect to factors with a significant difference are conducted.

**The Factor *Profiling Method*** Tables A.4(a), (b), and (c) show the MAPs with respect to each profiling method, the post-hoc analysis for the factor *Profiling Method*, and the effect size, respectively. Table A.4(a) presents the means and standard deviations of the three profiling methods. Table A.4(b) shows p-values of each pair. Since Table A.3 shows that the factor *Profiling Method* has the largest effect size, we further compute the effect size using Cohen's *d* for each pair shown in Table A.4(c). The result shows that CF-IDF and HCF-IDF are superior to LDA. In contrast, there is no significant difference between CF-IDF and HCF-IDF, although MAP of HCF-IDF is slightly higher than CF-IDF.

**The Factor *Publication Content*** Table A.5 shows the post-hoc analysis for the factor *Publication Content*. It indicates that the recommender system works better when All (i .e., full texts and titles) is taken into consideration for computing recommendations.

**The Factor *Profiling Method* × *Publication Content*** Table A.6 shows the results of ANOVA regarding the factor *Profiling Method* when a choice of the factor *Publication Content* is fixed and vice versa. We observe there are significant differences when a choice of the factor *Publication Content* is fixed and when CF-IDF is employed. Mendoza's test found a violation of sphericity in the factor

Table A.4: MAPs, Post-hoc analysis for the factor *Profiling Method* using Shaffer's MSRB procedure, and effect size.

**a) MAPs**

| **Choice** | **M** | **SD** |
|---|---|---|
| HCF-IDF | .65 | .33 |
| CF-IDF | .62 | .35 |
| LDA | .43 | .38 |

**b) Post-hoc analysis p-values**

|  | HCF-IDF | LDA |
|---|---|---|
| CF-IDF | .15 | **.00** |
| HCF-IDF |  | **.00** |

**c) Effect size using Cohen's $d$**

|  | HCF-IDF | LDA |
|---|---|---|
| CF-IDF | .09 | .52 |
| HCF-IDF |  | .62 |

Table A.5: MAPs and Post-hoc analysis for the factor *Publication Content* using Shaffer's MSRB procedure.

**a) MAPs**

| **Choice** | **M** | **SD** |
|---|---|---|
| All | .59 | .38 |
| Title | .55 | .35 |

**b) Post-hoc analysis p-values**

|  | Title |
|---|---|
| All | **.02** |

Table A.6: ANOVA for *Profiling Method* × *Publication Content* interaction

| **Factor** | **F** | $\eta^2$ | **p** |
|---|---|---|---|
| *Profiling Method* at Title | 23.99 | .20 | **.00** |
| *Profiling Method* at All | 36.35 | .30 | **.00** |
| *Publication Content* at CF-IDF | 14.69 | .12 | **.00** |
| *Publication Content* at HCF-IDF | 0.00 | .00 | .95 |
| *Publication Content* at LDA | 0.01 | .00 | .93 |

Table A.7: MAPs and Post-hoc analysis for the factor *Profiling Method* at Title using Shaffer's MSRB procedure.

**a) MAPs**

| **Choice** | **M** | **SD** |
|---|---|---|
| HCF-IDF | .65 | .33 |
| CF-IDF | .56 | .35 |
| LDA | .43 | .35 |

**b) Post-hoc analysis p-values**

| | HCF-IDF | LDA |
|---|---|---|
| CF-IDF | **.01** | **.00** |
| HCF-IDF | | **.00** |

*Profiling Method* when All is taken ($\chi^2(2) = 31.35$, $p = .00$). Thus, we run a one-way repeated-measure ANOVA with Greenhouse-Geisser correction of $\epsilon = .81$ for the second row in Table A.6. Subsequently, we conduct the post-hoc analyses for each factor with a significant difference. Table A.7 presents the post-hoc analysis when Title is employed. We see that HCF-IDF outperforms the others with significant differences. Table A.8 shows the post-hoc analysis when All is chosen for the factor *Publication Content*. Different from the result shown in Table A.7, CF-IDF performs slightly better than HCF-IDF, although there is no significant difference between them. Both CF-IDF and HCF-IDF demonstrate better recommendation performance than LDA. Table A.9 shows the post-hoc analysis of the factor *Publication Content* when CF-IDF is employed. It indicates that the strategies with CF-IDF and All significantly outperforms those with CF-IDF and Title.

Table A.8: MAPs and Post-hoc analysis for the factor *Profiling Method* at All using Shaffer's MSRB procedure.

**a) MAPs**

| **Choice** | **M** | **SD** |
|---|---|---|
| CF-IDF | .68 | .34 |
| HCF-IDF | .65 | .34 |
| LDA | .44 | .41 |

**b) Post-hoc analysis p-values**

| | HCF-IDF | LDA |
|---|---|---|
| CF-IDF | .21 | **.00** |
| HCF-IDF | | **.00** |

Table A.9: MAPs and Post-hoc analysis for the factor *Publication Content* at CF-IDF using Shaffer's MSRB procedure.

**a) MAPs**

| **Choice** | **M** | **SD** |
|---|---|---|
| All | .68 | .34 |
| Title | .56 | .35 |

**b) Post-hoc analysis p-values**

| | All |
|---|---|
| Title | **.00** |

## A.2 Precision

This section evaluates the recommendation performance using Precision, especially Precision@5 (P@5). Precision is computed as:

$$Precision@k = \frac{1}{k}\sum_{i=1}^{k} rel(i), \tag{A.2}$$

where $rel(k)$ returns 1 if the publication ranked at $i$ is interesting and 0 if not interesting. This section sets $k = 5$, since five publications are recommended by each strategy in the experiment. Using Precision@5, we first compare the twelve different strategies. Subsequently, we investigate the influence of the different experimental factors.

### A.2.1 Best Performing Strategy

Table A.10 shows Precision@5 of each strategy. For the statistical analyses, we first apply Mauchly's test and found a violation of sphericity in the strategies ($\chi^2(65) = 421.32$, $p = .00$). Subsequently, we run a one-way repeated-measure

Table A.10: Precision@5 (P@5) of the strategies in decreasing order. M and SD denote mean and standard deviation, respectively.

| | **Strategy** | | | **P@5** |
|---|---|---|---|---|
| | **Profiling Method** | **Decay Function** | **Content** | **M (SD)** |
| 1. | CF-IDF | Sliding Window | All | .59 (.33) |
| 2. | HCF-IDF | Sliding Window | All | .56 (.33) |
| 3. | HCF-IDF | Sliding Window | Title | .55 (.33) |
| 4. | HCF-IDF | Exponential | Title | .52 (.30) |
| 5. | CF-IDF | Exponential | All | .50 (.32) |
| 6. | HCF-IDF | Exponential | All | .48 (.30) |
| 7. | CF-IDF | Exponential | Title | .40 (.29) |
| 8. | CF-IDF | Sliding Window | Title | .39 (.27) |
| 9. | LDA | Exponential | Title | .37 (.31) |
| 10. | LDA | Sliding Window | Title | .34 (.31) |
| 11. | LDA | Exponential | All | .31 (.30) |
| 12. | LDA | Sliding Window | All | .27 (.33) |

ANOVA with a Greenhouse-Geisser correction of $\epsilon = .60$. It reveals a significant difference of the strategies ($F(6.62, 808.00) = 21.85$, $p = .00$). To assess the statistical significance of pair-wise differences between the twelve strategies, a post-hoc analysis is performed using Shaffer's MSRB procedure [Sha86]. The result of the post-hoc analysis is presented in Table A.11. The vertical and horizontal dimensions of the table show the eleven-by-eleven comparison of the twelve strategies. As one can see, we observe various significant differences between strategies (marked in bold font).

### A.2.2 Influence of the Three Experimental Factors

Subsequently, we analyze the results with respect to each factor with Precision@5. First, we apply Mendoza's test [Men80] which showed violations of sphericity against the factors *Profiling Method* ($\chi^2(2) = 13.92$, $p = .00$), *Profiling Method* × *Temporal Decay Function* ($\chi^2(2) = 19.64$, $p = .00$), and *Profiling Method* × *Publication Content* ($\chi^2(2) = 7.23$, $p = .03$). Thus, we run three-way repeated-measure ANOVA with a Greenhouse-Geisser correction of $\epsilon = .90$ for the factor *Profiling Method*, $\epsilon = .87$ for the factor *Profiling Method* × *Temporal Decay Function*, and $\epsilon = .95$ for the factor *Profiling Method* × *Publication Content*. Table A.12 shows the result of an ANOVA with F-ratio, $\eta^2$ and p-value. $\eta^2$ indicates the effect size of each factor. For all factors that make a significant difference, we conduct a post-hoc analysis using Shaffer's MSRB Procedure.

Table A.11: Post-hoc analysis of Precision@5 (P@5) with p-values for pairwise comparison over the strategies using Shaffer's MSRB procedure. The p-values are marked in bold font if $p < .05$, which indicates a significant difference between the two strategies. Strategies are sorted by P@5 as shown in Table A.10.

| | | | | All | Title | Title | All | All | Title | Title | Title | Title | All | All |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Sliding Window | Sliding Window | Exponential | Exponential | Exponential | Exponential | Sliding Window | Exponential | Sliding Window | Exponential | Sliding Window |
| | | | | HCF-IDF | HCF-IDF | HCF-IDF | CF-IDF | HCF-IDF | CF-IDF | CF-IDF | LDA | LDA | LDA | LDA |
| | | | | 2. | 3. | 4. | 5. | 6. | 7. | 8. | 9. | 10. | 11. | 12. |
| 1. | CF-IDF | Sliding Window | All | .99 | .99 | .60 | .09 | **.05** | **.00** | **.00** | **.00** | **.00** | **.00** | **.00** |
| 2. | HCF-IDF | Sliding Window | All | | .99 | .99 | .99 | .50 | **.00** | **.00** | **.00** | **.00** | **.00** | **.00** |
| 3. | HCF-IDF | Sliding Window | Title | | | .99 | .99 | .99 | **.00** | **.00** | **.00** | **.00** | **.00** | **.00** |
| 4. | HCF-IDF | Exponential | Title | | | | .99 | .99 | **.00** | **.00** | **.00** | **.00** | **.00** | **.00** |
| 5. | CF-IDF | Exponential | All | | | | | .99 | **.03** | **.01** | **.02** | **.00** | **.00** | **.00** |
| 6. | HCF-IDF | Exponential | All | | | | | | .09 | **.03** | **.03** | **.00** | **.00** | **.00** |
| 7. | CF-IDF | Exponential | Title | | | | | | | .99 | .99 | .99 | .26 | **.01** |
| 8. | CF-IDF | Sliding Window | Title | | | | | | | | .99 | .99 | .34 | **.02** |
| 9. | LDA | Exponential | Title | | | | | | | | | .99 | .99 | .09 |
| 10. | LDA | Sliding Window | Title | | | | | | | | | | .99 | .82 |
| 11. | LDA | Exponential | All | | | | | | | | | | | .99 |

Table A.12: Three-way repeated-measure ANOVA with Greenhouse-Geisser correction with F-ratio, $\eta^2$ and p-value for Precision@5.

| **Factor** | **F** | $\eta^2$ | **p** |
|---|---|---|---|
| *Profiling Method* | 54.24 | .42 | **.00** |
| *Temporal Decay Function* | 1.75 | .00 | .19 |
| *Publication Content* | 3.23 | .04 | .08 |
| *Profiling Method* × *Temporal Decay Function* | 6.32 | .01 | **.00** |
| *Profiling Method* × *Publication Content* | 20.53 | .02 | **.00** |
| *Temporal Decay Function* × *Publication Content* | 7.13 | .00 | **.01** |
| *Profiling Method* × *Temporal Decay Function* × *Publication Content* | 2.61 | .03 | .07 |

Table A.13: Precision@5, Post-hoc analysis for the factor *Profiling Method* using Shaffer's MSRB procedure, and effect size.

**a) Precision@5**

| **Choice** | **M** | **SD** |
|---|---|---|
| HCF-IDF | .53 | .31 |
| CF-IDF | .47 | .31 |
| LDA | .32 | .31 |

**b) Post-hoc analysis p-values**

| | HCF-IDF | LDA |
|---|---|---|
| CF-IDF | **.00** | **.00** |
| HCF-IDF | | **.00** |

**c) Effect size using Cohen's *d***

| | HCF-IDF | LDA |
|---|---|---|
| CF-IDF | .09 | .52 |
| HCF-IDF | | .62 |

**The Factor *Profiling Method*** Tables A.13(a), (b), and (c) show the Precision@5, the post-hoc analysis for the factor *Profiling Method*, and the effect size, respectively. Table A.13(a) presents the means and standard deviations of the three profiling methods. Table A.13(b) shows p-values of each pair. Since Table A.12 shows that this factor has the largest effect size, we further compute the effect size using Cohen's $d$ for each pair shown in Table A.13(c). There are significant differences between all pairs of the three profiling methods and among the three profiling methods HCF-IDF performs best.

**The Factor *Profiling Method* × *Temporal Decay Function*** Table A.14 shows the results of ANOVA regarding the *Profiling Method* when a choice of the *Temporal Decay Function* is fixed and vice versa. There are significant differences when the choice of the factor *Temporal Decay Function* is fixed. In both temporal

Table A.14: ANOVA for *Profiling Method* × *Temporal Decay Function* interaction

| **Factor** | **F** | $\eta^2$ | **p** |
|---|---|---|---|
| *Profiling Method* at Sliding Window | 52.98 | .20 | **.00** |
| *Profiling Method* at Exponential | 22.52 | .30 | **.00** |
| *Temporal Decay Function* at CF-IDF | 5.44 | .12 | **.02** |
| *Temporal Decay Function* at HCF-IDF | 3.25 | .00 | .07 |
| *Temporal Decay Function* at LDA | 6.75 | .00 | **.01** |

Table A.15: ANOVA for *Profiling Method* × *Publication Content* interaction

| **Factor** | **F** | $\eta^2$ | **p** |
|---|---|---|---|
| *Profiling Method* at Title | 23.37 | .20 | **.00** |
| *Profiling Method* at All | 56.54 | .30 | **.00** |
| *Publication Content* at CF-IDF | 33.39 | .12 | **.00** |
| *Publication Content* at HCF-IDF | 0.44 | .00 | .51 |
| *Publication Content* at LDA | 4.68 | .00 | **.03** |

decay functions, all pairs of the three profiling methods show significant differences. Specifically, HCF-IDF performs best, followed by CF-IDF and LDA. When CF-IDF is employed, Sliding Window makes significantly better recommendations than Exponential ($F(1, 122) = 5.44$, $p = .02$). In contrast, when LDA is employed, Exponential performs significantly better than Sliding Window ($F(1, 122) = 6.75$, $p = .01$). The factor *Temporal Decay Function* does not make difference on the recommendation performance when HCF-IDF is employed.

**The Factor *Profiling Method* × *Publication Content*** Table A.15 shows the results of ANOVA regarding the factor *Profiling Method* when a choice of the *Publication Content* is fixed and vice versa. When the choice of the *Publication Content* is Title, HCF-IDF performs best and significantly better than both CF-IDF and LDA. There is no significant difference between CF-IDF and LDA. When the choice of the *Publication Content* is All, HCF-IDF performs best. But, there is no significant difference between CF-IDF and HCF-IDF and both profiling methods are significantly superior to LDA. When CF-IDF is employed, All is the better choice than Title. In contrast, Title performs better than All, when LDA is employed.

**The Factor *Temporal Decay Function* × *Publication Content*** Table A.16 shows the results of ANOVA regarding the factor *Temporal Decay Function* when a choice of the factor *Publication Content* is fixed and vice versa. When All is chosen for the factor *Publication Content*, Sliding Window is the better

Table A.16: ANOVA for *Temporal Decay Function* × *Publication Content* interaction

| **Factor** | **F** | $\eta^2$ | **p** |
|---|---|---|---|
| *Temporal Decay Function* at Title | 0.08 | .20 | .78 |
| *Temporal Decay Function* at All | 4.99 | .30 | **.03** |
| *Publication Content* at Sliding Window | 8.74 | .12 | **.00** |
| *Publication Content* at Exponential | 0.00 | .00 | .97 |

temporal decay function. When Sliding Window is employed in the strategies, the strategies with All is significantly better than those with Title.

## A.3 Mean Reciprocal Rank

In this section, we evaluate the recommendation performance by computing Mean Reciprocal Rank (MRR). Reciprocal Rank is defined as:

$$RR = \frac{1}{rank_{first}}, \tag{A.3}$$

where $rank_{first}$ indicates the rank position of the first publication which is evaluated as interesting. Mean Reciprocal Rank (MRR) is the mean average of the Reciprocal Rank of all subjects. If there is no relevant publication in the recommendation list, RR outputs 0. Using MRR, we first compare the twelve different strategies. Subsequently, we investigate the influence of the different experimental factors.

### A.3.1 Best Performing Strategy

Table A.17 shows the MRR of each strategies. The order of the strategies are different from rankscores shown in Table 5.2. For the statistical analyses, we first applied Mauchly's test and found a violation of sphericity in the strategies ($\chi^2(65) = 308.70$, $p = .00$). Subsequently, we ran a one-way repeated-measure ANOVA with a Greenhouse-Geisser correction of $\epsilon = .67$. It revealed a significant difference of the strategies' MRRs ($F(0.18, 2.53) = 14.40$, $p = .00$). To assess the statistical significance of pair-wise differences between the twelve strategies, a post-hoc analysis was performed using Shaffer's MSRB procedure [Sha86]. The result of the post-hoc analysis is presented in Table A.18. The vertical and horizontal dimensions of the table show the eleven-by-eleven comparison of the

Table A.17: Mean Reciprocal Rank (MRR) of the strategies in decreasing order. M and SD denote mean and standard deviation, respectively.

| | **Strategy** | | | **MRR** |
|---|---|---|---|---|
| | **Profiling Method** | **Decay Function** | **Content** | **M (SD)** |
| 1 | CF-IDF | Sliding Window | All | .73 (.35) |
| 2 | CF-IDF | Exponential | All | .69 (.39) |
| 3 | HCF-IDF | Exponential | All | .68 (.37) |
| 4 | HCF-IDF | Exponential | Title | .68 (.37) |
| 5 | HCF-IDF | Sliding Window | Title | .67 (.38) |
| 6 | HCF-IDF | Sliding Window | All | .67 (.37) |
| 7 | CF-IDF | Exponential | Title | .61 (.39) |
| 8 | CF-IDF | Sliding Window | Title | .59 (.39) |
| 9 | LDA | Exponential | All | .50 (.43) |
| 10 | LDA | Exponential | Title | .43 (.37) |
| 11 | LDA | Sliding Window | Title | .42 (.38) |
| 12 | LDA | Sliding Window | All | .41 (.44) |

twelve strategies. As one can see, we observe various significant differences between strategies (marked in bold font).

### A.3.2 Influence of the Three Experimental Factors

Subsequently, we analyze the results with respect to each factor with MRR. First, we apply Mendoza's test [Men80], which shows violations of sphericity against the factors *Profiling Method* × *Temporal Decay Function* ($\chi^2(2) = 8.16$, $p = .02$), and *Profiling Method* × *Publication Content* ($\chi^2(2) = 8.85$, $p = .01$). Thus, we run three-way repeated-measure ANOVA with a Greenhouse-Geisser correction of $\epsilon = .94$ for *Profiling Method* × *Temporal Decay Function*, and $\epsilon = .93$ for *Profiling Method* × *Publication Content*. Table A.19 shows the results of an ANOVA with F-ratio, $\eta^2$ and p-value. The analysis revealed significant differences only in the two factors *Profiling Method* and *Publication Content*.

**The Factor *Profiling Method*** Tables A.20(a), (b), and (c) show the MRRs, the post-hoc analysis for the factor *Profiling Method*, and the effect size, respectively. Table A.20(a) presents the means and standard deviations of the three profiling methods. Table A.20(b) shows p-values of each pair. Since Table A.19 shows that this factor has the largest effect size, we further compute the effect size using Cohen's $d$ for each pair shown in Table A.20(c). The result indicates

Table A.18: Post-hoc analysis of Mean Reciprocal Rank (MRR) with p-values for pairwise comparison over the strategies using Shaffer's MSRB procedure. The p-values are marked in bold font if $p < .05$, which indicates a significant difference between the two strategies. Strategies are sorted by MRR as shown in Table A.17.

| | | | | All | All | Title | Title | All | Title | Title | All | Title | Title | All |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Exponential | Exponential | Exponential | Sliding Window | Sliding Window | Exponential | Sliding Window | Exponential | Exponential | Sliding Window | Sliding Window |
| | | | | CF-IDF | HCF-IDF | HCF-IDF | HCF-IDF | HCF-IDF | CF-IDF | CF-IDF | LDA | LDA | LDA | LDA |
| | | | | 2. | 3. | 4. | 5. | 6. | 7. | 8. | 9. | 10. | 11. | 12. |
| 1. | CF-IDF | Sliding Window | All | .99 | .99 | .99 | .99 | .99 | .12 | **.03** | **.00** | **.00** | **.00** | **.00** |
| 2. | CF-IDF | Exponential | All | | .99 | .99 | .99 | .99 | .99 | .99 | **.01** | **.00** | **.00** | **.00** |
| 3. | HCF-IDF | Exponential | All | | | .99 | .99 | .99 | .99 | .99 | **.01** | **.00** | **.00** | **.01** |
| 4. | HCF-IDF | Exponential | Title | | | | .99 | .99 | .99 | .99 | **.01** | **.00** | **.00** | **.00** |
| 5. | HCF-IDF | Sliding Window | Title | | | | | .99 | .99 | .99 | **.01** | **.00** | **.00** | **.00** |
| 6. | HCF-IDF | Sliding Window | All | | | | | | .99 | .99 | **.01** | **.00** | **.00** | **.01** |
| 7. | CF-IDF | Exponential | Title | | | | | | | .99 | .94 | **.00** | **.00** | **.00** |
| 8. | CF-IDF | Sliding Window | Title | | | | | | | | .99 | **.00** | **.00** | **.02** |
| 9. | LDA | Exponential | All | | | | | | | | | .99 | .99 | .58 |
| 10. | LDA | Exponential | Title | | | | | | | | | | .99 | .99 |
| 11. | LDA | Sliding Window | Title | | | | | | | | | | | .99 |

Table A.19: Three-way repeated-measure ANOVA with Greenhouse-Geisser correction with F-ratio, $\eta^2$ and p-value for MRR.

| **Factor** | **F** | $\eta^2$ | **p** |
|---|---|---|---|
| *Profiling Method* | 50.65 | .42 | **.00** |
| *Temporal Decay Function* | 0.56 | .00 | .45 |
| *Publication Content* | 5.10 | .04 | **.03** |
| *Profiling Method* × *Temporal Decay Function* | 1.28 | .01 | .28 |
| *Profiling Method* × *Publication Content* | 2.83 | .02 | .06 |
| *Temporal Decay Function* × *Publication Content* | 0.13 | .00 | .72 |
| *Profiling Method* × *Temporal Decay Function* × *Publication Content* | 2.33 | .02 | .10 |

Table A.20: MRRs, Post-hoc analysis for the factor *Profiling Method* using Shaffer's MSRB procedure, and effect size.

**a) MRRs**

| **Choice** | **M** | **SD** |
|---|---|---|
| HCF-IDF | .68 | .38 |
| CF-IDF | .66 | .37 |
| LDA | .44 | .41 |

**b) Post-hoc analysis p-values**

| | HCF-IDF | LDA |
|---|---|---|
| CF-IDF | .34 | **.00** |
| HCF-IDF | | **.00** |

**c) Effect size using Cohen's *d***

| | HCF-IDF | LDA |
|---|---|---|
| CF-IDF | .05 | .56 |
| HCF-IDF | | .61 |

that both CF-IDF and HCF-IDF outperform LDA. On the other hand, CF-IDF and HCF-IDF are competitive each other.

**The Factor *Publication Content*** Table A.21 shows the post-hoc analysis for the factor *Publication Content*. It indicates that generally the recommender system works better when full texts are available.

## A.4 Normalized Discounted Cumulative Gain

In this section, we evaluate the recommendation performance by Normalized Discounted Cumulative Gain (nDCG). Discounted Cumulative Gain (DCG) is

Table A.21: MRRs and Post-hoc analysis for the factor *Publication Content* using Shaffer's MSRB procedure.

**a) MRRs**

| Choice | M | SD |
|---|---|---|
| All | .61 | .41 |
| Title | .57 | .39 |

**b) Post-hoc analysis p-values**

| | All |
|---|---|
| Title | **.03** |

calculated as:

$$DCG = \sum_{i=1}^{k} \frac{2^{rel(i)} - 1}{\log_2 i}, \tag{A.4}$$

where $rel(k)$ returns 1 if the publication ranked at $i$ is interesting and 0 if not interesting. Similar to rankscore, the items ranked at higher positions have a larger influence on output score. First, we compare the twelve different strategies using this metric. Subsequently, we investigate the influence of the different experimental factors.

## A.4.1 Best Performing Strategy

Table A.22 shows the Normalized Discounted Cumulative Gain (nDCG) of the twelve strategies. The order of the strategies is identical with rankscores shown in Table 5.2. For the statistical analyses, we first apply Mauchly's test and found a violation of sphericity in the strategies ($\chi^2(65) = 424.00$, $p = .00$). Subsequently, we run a one-way repeated-measure ANOVA with a Greenhouse-Geisser correction of $\epsilon = .61$. It reveals a significant difference of the strategies' nDCG ($F(6.69, 816.37) = 21.16$, $p = .00$). To assess pair-wise differences between the twelve strategies, a post-hoc analysis is performed using Shaffer's MSRB procedure [Sha86]. The result of the post-hoc analysis is presented in Table A.23. The vertical and horizontal dimensions of the table show the eleven-by-eleven comparison of the twelve strategies. As one can see, we observe various significant differences between strategies (marked in bold font).

## A.4.2 Influence of the Three Experimental factors

Subsequently, we investigate the influence of the different experimental factors. First, we apply Mendoza's test [Men80] which shows violations of sphericity against the factors *Profiling Method* ($\chi^2(2) = 11.29$, $p = .00$), *Profiling Method*

Table A.22: nDCGs of the strategies in decreasing order. M and SD denote mean and standard deviation, respectively.

| | **Strategy** | | | **nDCG** |
|---|---|---|---|---|
| | **Profiling Method** | **Decay Function** | **Publication Content** | **M (SD)** |
| 1. | CF-IDF | Sliding Window | All | .59 (.33) |
| 2. | HCF-IDF | Sliding Window | All | .56 (.34) |
| 3. | HCF-IDF | Sliding Window | Title | .55 (.33) |
| 4. | HCF-IDF | Exponential | Title | .52 (.30) |
| 5. | CF-IDF | Exponential | All | .52 (.32) |
| 6. | HCF-IDF | Exponential | All | .50 (.30) |
| 7. | CF-IDF | Exponential | Title | .41 (.30) |
| 8. | CF-IDF | Sliding Window | Title | .40 (.27) |
| 9. | LDA | Exponential | Title | .34 (.31) |
| 10. | LDA | Sliding Window | Title | .32 (.31) |
| 11. | LDA | Exponential | All | .32 (.31) |
| 12. | LDA | Sliding Window | All | .28 (.33) |

× *Temporal Decay Function* ($\chi^2(2) = 18.90$, $p = .00$), and *Profiling Method* × *Publication Content* ($\chi^2(2) = 8.61$, $p = .01$). Thus, we run three-way repeated-measure ANOVA with a Greenhouse-Geisser correction of $\epsilon = .92$ for the factor *Profiling Method*, $\epsilon = .87$ for the factor *Profiling Method* × *Temporal Decay Function*, and $\epsilon = .94$ for the factor *Profiling Method* × *Publication Content*. Table A.24 shows the results of the ANOVA. $\eta^2$ indicates the effect size of each factor. For all the factors that make significant difference, we conduct a post-hoc analysis using Shaffer's MSRB Procedure.

**The Factor *Profiling Method*** Tables A.25(a), (b), and (c) show the nDCGs, the post-hoc analysis for the factor *Profiling Method*, and the effect size, respectively. Table A.25(a) presents the means and standard deviations of the three profiling methods. Table A.25(b) shows p-values of each pair. Since Table A.24 shows that the factor *Profiling Method* has the largest effect size, we further compute the effect size using Cohen's $d$ for each pair shown in Table A.25(c). The result shows that HCF-IDF is the best profiling method, followed by CF-IDF and LDA.

**The Factor *Publication Content*** Table A.26 shows the post-hoc analysis for the factor *Publication Content*. It indicates that the recommender system works better when All (i .e., full texts and titles) is taken into consideration.

Table A.23: Post-hoc analysis of normalized Discounted Cumulative Gain (nDCG) with p-values for pairwise comparison over the twelve strategies using Shaffer's MSRB procedure. The p-values are marked in bold font if $p < .05$, which indicates a significant difference between the two strategies. Strategies are sorted by nDCG as shown in Table A.22.

| | | | | All | Title | Title | All | All | Title | Title | Title | Title | All | All |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Sliding Window | Sliding Window | Exponential | Exponential | Exponential | Exponential | Sliding Window | Exponential | Sliding Window | Exponential | Sliding Window |
| | | | | HCF-IDF | HCF-IDF | HCF-IDF | CF-IDF | HCF-IDF | CF-IDF | CF-IDF | LDA | LDA | LDA | LDA |
| | | | | 2. | 3. | 4. | 5. | 6. | 7. | 8. | 9. | 10. | 11. | 12. |
| 1. | CF-IDF | Sliding Window | All | .99 | .99 | .85 | .41 | .22 | **.00** | **.00** | **.00** | **.00** | **.00** | **.00** |
| 2. | HCF-IDF | Sliding Window | All | | .99 | .99 | .99 | .99 | **.00** | **.00** | **.00** | **.00** | **.00** | **.00** |
| 3. | HCF-IDF | Sliding Window | Title | | | .99 | .99 | .99 | **.01** | **.00** | **.00** | **.00** | **.00** | **.00** |
| 4. | HCF-IDF | Exponential | Title | | | | .99 | .99 | **.01** | **.00** | **.00** | **.00** | **.00** | **.00** |
| 5. | CF-IDF | Exponential | All | | | | | .99 | .05 | **.01** | **.00** | **.00** | **.00** | **.00** |
| 6. | HCF-IDF | Exponential | All | | | | | | .17 | **.03** | **.00** | **.00** | **.00** | **.00** |
| 7. | CF-IDF | Exponential | Title | | | | | | | .99 | .85 | .18 | .34 | **.01** |
| 8. | CF-IDF | Sliding Window | Title | | | | | | | | .99 | .41 | .77 | **.05** |
| 9. | LDA | Exponential | Title | | | | | | | | | .99 | .99 | .99 |
| 10. | LDA | Sliding Window | Title | | | | | | | | | | .99 | .99 |
| 11. | LDA | Exponential | All | | | | | | | | | | | .98 |

Table A.24: Three-way repeated-measure ANOVA with Greenhouse-Geisser correction with F-ratio, $\eta^2$, and p-value for nDCG.

| **Factor** | **F** | $\eta^2$ | **p** |
|---|---|---|---|
| *Profiling Method* | 58.42 | .48 | **.00** |
| *Temporal Decay Function* | 0.80 | .01 | .37 |
| *Publication Content* | 6.33 | .05 | **.01** |
| *Profiling Method* × *Temporal Decay Function* | 3.81 | .03 | **.03** |
| *Profiling Method* × *Publication Content* | 14.54 | .12 | **.00** |
| *Temporal Decay Function* × *Publication Content* | 3.57 | .03 | .06 |
| *Profiling Method* × *Temporal Decay Function* × *Publication Content* | 3.09 | .03 | **.05** |

Table A.25: nDCGs, Post-hoc analysis for the factor *Profiling Method* using Shaffer's MSRB procedure, and effect size.

**a) nDCGs**

| **Choice** | **M** | **SD** |
|---|---|---|
| HCF-IDF | .53 | .32 |
| CF-IDF | .48 | .32 |
| LDA | .32 | .32 |

**b) Post-hoc analysis p-values**

| | HCF-IDF | LDA |
|---|---|---|
| CF-IDF | **.00** | **.00** |
| HCF-IDF | | **.00** |

**c) Effect size using Cohen's *d***

| | HCF-IDF | LDA |
|---|---|---|
| CF-IDF | .16 | .50 |
| HCF-IDF | | .65 |

Table A.26: nDCGs and Post-hoc analysis for the factor *Publication Content* using Shaffer's MSRB procedure.

**a) nDCGs**

| **Choice** | **M** | **SD** |
|---|---|---|
| All | .46 | .34 |
| Title | .42 | .31 |

**b) Post-hoc analysis p-values**

| | Title |
|---|---|
| All | **.01** |

Table A.27: ANOVA for *Profiling Method* × *Temporal Decay Function* interaction

| **Factor** | **F** | $\eta^2$ | **p** |
|---|---|---|---|
| *Profiling Method* at Sliding Window | 50.59 | .41 | **.00** |
| *Profiling Method* at Exponential | 27.92 | .23 | **.00** |
| *Temporal Decay Function* at CF-IDF | 2.79 | .02 | .10 |
| *Temporal Decay Function* at HCF-IDF | 1.78 | .01 | .18 |
| *Temporal Decay Function* at LDA | 4.90 | .04 | **.03** |

Table A.28: nDCGs and Post-hoc analysis for the factor *Profiling Method* at Sliding Window using Shaffer's MSRB procedure.

**a) nDCGs**

| **Choice** | **M** | **SD** |
|---|---|---|
| HCF-IDF | .55 | .33 |
| CF-IDF | .50 | .32 |
| LDA | .30 | .32 |

**b) Post-hoc analysis p-values**

| | HCF-IDF | LDA |
|---|---|---|
| CF-IDF | **.02** | **.00** |
| HCF-IDF | ▨ | **.00** |

**The Factor *Profiling Method* × *Temporal Decay Function*** Table A.27 shows the results of ANOVA regarding the factor *Profiling Method* when a choice of the factor *Temporal Decay Function* is fixed and vice versa. Mendoza's test finds a violation of sphericity in the factor *Profiling Method* when Sliding Window is used ($\chi^2(2) = 7.55$, $p = .02$) and Exponential is used ($\chi^2(2) = 10.74$, $p = .00$). Thus, we run a one-way repeated-measure ANOVA with Greenhouse-Geisser correction of $\eta = .94$ for the first row and $\eta = .92$ for the second row in Table A.27. We also observe significant differences in the factor *Temporal Decay Function* when LDA is employed. The post-hoc analyses of them are shown in Tables A.28, A.29, and A.30, respectively. In Table A.28 and Table A.29, a choice of the factor *Temporal Decay Function* is fixed. Table A.30 shows the post-hoc analysis of the factor *Temporal Decay Function* when LDA is employed. It indicates Exponential performs better than Sliding Window when using LDA.

**The Factor *Profiling Method* × *Publication Content*** Table A.31 shows the results of ANOVA regarding the factor *Profiling Method* when a choice of the factor *Publication Content* is fixed and vice versa. We observe there are significant differences when a choice of the factor *Publication Content* is fixed and CF-IDF is employed. Mendoza's test found a violation of sphericity in the

Table A.29: nDCGs and Post-hoc analysis for the factor *Profiling Method* at Exponential using Shaffer's MSRB procedure.

**a) nDCGs**

| Choice | M | SD |
|---|---|---|
| HCF-IDF | .51 | .30 |
| CF-IDF | .46 | .31 |
| LDA | .34 | .31 |

**b) Post-hoc analysis p-values**

| | HCF-IDF | LDA |
|---|---|---|
| CF-IDF | **.03** | **.00** |
| HCF-IDF | ▨ | **.00** |

Table A.30: nDCGs and Post-hoc analysis for the factor *Temporal Decay Function* at LDA using Shaffer's MSRB procedure.

**a) nDCGs**

| Choice | M | SD |
|---|---|---|
| Exponential | .34 | .31 |
| Sliding Window | .30 | .32 |

**b) Post-hoc analysis p-value**

| | Exponential |
|---|---|
| Sliding Window | **.03** |

factor *Profiling Method* when All is taken ($\chi^2(2) = 24.64$, $p = .00$). Thus, we run a one-way repeated-measure ANOVA with Greenhouse-Geisser correction of $\eta = .84$ for the second row in Table A.31. Table A.32 presents the post-hoc analysis when Title is employed. We see that HCF-IDF outperforms others with significant differences. Table A.33 shows the post-hoc analysis when All is chosen for the factor *Publication Content*. While there is no significant difference between CF-IDF and HCF-IDF in Table A.33, HCF-IDF outperforms CF-IDF when only titles of publications are available according to Table A.32. Table A.34 shows the post-hoc analysis of the factor *Publication Content* when CF-IDF is employed. It indicates that the strategies with CF-IDF and All significantly outperforms those with CF-IDF and Title.

**The Factor *Temporal Decay Function* × *Publication Content*** Table A.35 shows the results of ANOVA regarding the factor *Temporal Decay Function* when a choice of the factor *Publication Content* is fixed and vice versa. According to Table A.35, there is a significant difference among the factor *Publication Content*, when Sliding Window is used. The nDCGs and post-hoc analysis of

Table A.31: ANOVA for *Profiling Method* × *Publication Content* interaction

| **Factor** | **F** | $\eta^2$ | **p** |
|---|---|---|---|
| *Profiling Method* at Title | 26.61 | .22 | **.00** |
| *Profiling Method* at All | 52.51 | .43 | **.00** |
| *Publication Content* at CF-IDF | 30.81 | .25 | **.00** |
| *Publication Content* at HCF-IDF | 0.31 | .00 | .58 |
| *Publication Content* at LDA | 0.94 | .01 | .33 |

Table A.32: nDCGs and Post-hoc analysis for the factor *Profiling Method* at Title using Shaffer's MSRB procedure.

**a) nDCGs**

| **Choice** | **M** | **SD** |
|---|---|---|
| HCF-IDF | .53 | .32 |
| CF-IDF | .41 | .29 |
| LDA | .33 | .31 |

**b) Post-hoc analysis p-values**

| | HCF-IDF | LDA |
|---|---|---|
| CF-IDF | **.00** | **.01** |
| HCF-IDF | | **.00** |

Table A.33: nDCG and Post-hoc analysis for the factor *Profiling Method* at All using Shaffer's MSRB procedure.

**a) nDCGs**

| **Choice** | **M** | **SD** |
|---|---|---|
| CF-IDF | .56 | .33 |
| HCF-IDF | .53 | .34 |
| LDA | .30 | .32 |

**b) Post-hoc analysis p-values**

| | HCF-IDF | LDA |
|---|---|---|
| CF-IDF | .18 | **.00** |
| HCF-IDF | | **.00** |

Table A.34: nDCGs and Post-hoc analysis for the factor *Publication Content* at CF-IDF using Shaffer's MSRB procedure.

**a) MAPs**

| **Choice** | **M** | **SD** |
|---|---|---|
| All | .56 | .33 |
| Title | .41 | .29 |

**b) Post-hoc analysis p-values**

| | All |
|---|---|
| Title | **.00** |

Table A.35: ANOVA for *Temporal Decay Function* × *Publication Content* interaction

| **Factor** | **F** | $\eta^2$ | **p** |
|---|---|---|---|
| *Temporal Decay Function* at Title | 0.06 | .00 | .81 |
| *Temporal Decay Function* at All | 2.28 | .02 | .13 |
| *Publication Content* at Sliding Window | 9.96 | .08 | **.00** |
| *Publication Content* at Exponential | 1.19 | .01 | .28 |

Table A.36: nDCGs and Post-hoc analysis for the factor *Publication Content* at Sliding Window using Shaffer's MSRB procedure.

**a) nDCGs**

| **Choice** | **M** | **SD** |
|---|---|---|
| All | .48 | .36 |
| Title | .42 | .32 |

**b) Post-hoc analysis p-value**

| | All |
|---|---|
| Title | **.00** |

it are shown in Tables A.36(a) and (b). It indicates that All significantly enhances the performance of the recommender system when Sliding Window is used.

## A.5 Demographic Factor

While Section 5.4.3 describes the demographic factors that have an influence on the recommendation performance, this section details the other demographic factors (i .e., age, major, years of profession, and employment type). For each of these demographic factor, we first apply Mendoza's test. Subsequently, we conduct a mixed ANOVA test with one between subject factor (i. e., demographic factor) and one within subject factor (i. e., strategy), adjusted by Green-house-Geisser's epsilon. In addition, we provide the post-hoc analyses. However, we omit the post-hoc analysis of the factor strategy for the sake of brevity, because it is not different from the result of the one-way repeated-measure ANOVA shown in Table 5.3.

**Age** On average, subjects are 32.90 years old (SD: 7.36). We divide subjects into three groups for an ANOVA (group 1: subjects who are $> 29$ years old ($n = 42$), group 2: $<= 29$ and $> 38$ years old ($n = 49$), group 3: $<= 38$ years old ($n = 32$)). We set those thresholds to make three groups have the almost same number of subjects. Mendoza's test found a violation of sphericity in the recommendation

Table A.37: Mixed ANOVA with a between subject factor *Age* and a within subject factor *Strategy* Greenhouse-Geisser correction with F-ratio, effect size $\eta^2$, and p-value.

| **Factor** | **F** | $\eta^2$ | **p** |
|---|---|---|---|
| Age | 2.06 | .03 | .13 |
| Strategy | 14.82 | .12 | **.00** |
| Age × Strategy | 0.69 | .01 | .77 |

Table A.38: Mixed ANOVA with a between subject factor *Major* and a within subject factor *Strategy* Greenhouse-Geisser correction with F-ratio, effect size $\eta^2$, and p-value.

| **Factor** | **F** | $\eta^2$ | **p** |
|---|---|---|---|
| Major | 0.01 | .00 | .94 |
| Strategy | 16.41 | .14 | **.00** |
| Major × Strategy | 1.73 | .01 | .10 |

strategies ($\chi^2(197) = 504.35$, $p = .00$). Table A.37 shows the result of an ANOVA with a Greenhouse-Geisser correction of $\epsilon = .60$. It indicates that the age of subjects has no influence on the performance of the different recommendation strategies.

**Major** In the experiment, subjects provide information about their majors. We manually classify subjects into the two groups: subjects whose major is economics ($n = 92$) and others ($n = 31$). Mendoza's test finds a violation of sphericity in the recommendation strategies for these two groups ($\chi^2(131) = 466.90$, $p = .00$). Table A.38 shows the result of an ANOVA with a Greenhouse-Geisser correction of $\epsilon = .60$. It indicates that the major of subjects has no influence on the performance of the different recommendation strategies.

**Years of Profession** On average, subjects work in their fields for 7.85 years (SD: 6.85). We divide subjects into three groups for an ANOVA (group 1: subjects who work for $> 5$ years ($n = 44$), group 2: $<= 5$ and $> 10$ years ($n = 34$), group 3: $<= 10$ years ($n = 44$)). We set those thresholds to make three groups have the almost same number of subjects. Mendoza's test reveals a violation of sphericity in the recommendation strategies ($\chi^2(197) = 541.67$, $p = .00$). Table A.39 shows the result of an ANOVA with a Greenhouse-Geisser correction of $\epsilon = .60$. It indicates that how long subjects have worked in their fields has no influence on the performance of the different recommendation strategies.

Table A.39: Mixed ANOVA with a between subject factor *Years of Profession* and a within subject factor *Strategy* Greenhouse-Geisser correction with F-ratio, effect size $\eta^2$, and p-value.

| **Factor** | **F** | $\eta^2$ | **p** |
|---|---|---|---|
| Years of Profession | 0.13 | .00 | .88 |
| Strategy | 21.70 | .18 | **.00** |
| Years of Profession × Strategy | 0.80 | .01 | .66 |

Table A.40: Mixed ANOVA with a between subject factor *Employment Type* and a within subject factor *Strategy* Greenhouse-Geisser correction with F-ratio, effect size $\eta^2$, and p-value.

| **Factor** | **F** | $\eta^2$ | **p** |
|---|---|---|---|
| Employment Type | 0.35 | .00 | .55 |
| Strategy | 18.05 | .15 | **.00** |
| Employment Type × Strategy | 0.97 | .01 | .45 |

**Employment Type** We have subjects who work in academia ($n = 83$) and industry ($n = 40$). Mendoza's test finds a violation of sphericity in the recommendation strategies ($\chi^2(131) = 472.14$, $p = .00$). Table A.40 shows the result of an ANOVA with a Greenhouse-Geisser correction of $\epsilon = .60$. It indicates that the employment type of subjects has no influence on the performance of the different recommendation strategies.