NATIONAL UNIVERSITY OF IRELAND GALWAY

DOCTORAL THESIS

Semantics-Aware User Modeling and Recommender Systems in Online Social Networks

---

Author:
Guangyuan Piao

Supervisor:
Dr. John G. Breslin

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Insight Centre for Data Analytics
Data Science Institute
College of Engineering and Informatics

August 8, 2018

Declaration of Authorship

I, Guangyuan Piao, declare that this thesis titled, "Semantics-Aware User Modeling and Recommender Systems in Online Social Networks" and the work presented in it are my own. I confirm that:

- I have not obtained a degree in this University or elsewhere on the basis of any of this work.
- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.
- The work reported in this thesis was supported by Science Foundation Ireland (SFI) and SAP Ireland under Grant Number SFI/12/RC/2289 (Insight Centre for Data Analytics).

Signed:

Date:

"A man is not old as long as he is seeking something. A man is not old until regrets take the place of dreams."

Jean Rostand

Abstract

Guangyuan Piao

Semantics-Aware User Modeling and Recommender Systems in Online Social Networks

The popularity of Online Social Networks (OSNs) has rapidly increased over the past few years. User modeling (including creating user interest profiles) and recommendation approaches in OSNs are important methods to deal with cold-start problems inside or outside those OSNs and to cope with the problem of information overload. Recently, semantics-aware techniques such as *top-down* approaches which explore external knowledge sources such as DBpedia and *bottom-up* approaches which learn latent semantic representations for users/items using factorization or embedding approaches have received great attention in the domain of user modeling and recommender systems.

The aims of this study were to propose semantics-aware approaches for user modeling and recommending items in OSNs. For *active* users who consistently generate content, various user modeling dimensions such as the temporal dynamics and semantics of user interests, and a comprehensive user modeling strategy considering those dimensions, have been investigated. For *passive* users who only follow other users in OSNs without generating content, various types of information about their followees (the users they follow in OSNs) have been investigated. Furthermore, this thesis investigates semantics-aware recommendation approaches based on semantic information from knowledge graphs (KGs) such as DBpedia, and proposes semantic similarity/distance measures and factorization approaches in the context of different recommendation scenarios such as a cold start.

The experimental results show that the strategy for *representing user interests* plays the most important role followed by the *temporal dynamics* of user interests in user modeling for active users. For passive users, the results show that both *biographies* and *list memberships* of followees provide useful information for inferring user interest profiles, and the profiles inferred based

on this information outperform the ones inferred based on the information from the tweets or account names of followees.

mLDSD, a proposed semantic similarity measure with a global normalization strategy outperforms other semantic similarity measures in the context of a cold-start scenario for item recommendations. When there is plenty of feedback from users, *LODFM*, a proposed factorization approach exploring lightweight DBpedia features outperforms other state-of-the-art methods significantly in two different domains. As the *incompleteness* of KGs had not been considered for semantics-aware recommendations in the literature, we further investigated transfer learning between item recommendations and knowledge graph completion. The results showed that considering the incompleteness of a KG can further improve the performance when compared to *LODFM*, and performs better than other baselines. In addition, the results show that exploiting *user-item interaction histories* also improves the performance of completing the KG with regard to the domain of items, which has not been investigated before.

Acknowledgements

First and foremost, I would like to thank my Ph.D. supervisor Dr. John G. Breslin who gave me the opportunity to start my Ph.D. in the Unit for Social Semantics at Insight Centre for Data Analytics. I appreciate his supervision and the freedom of research given to me along the way, which allows me to explore research topics that I would like to explore and makes me become an independent researcher. I want to express my sincere gratitude for his continuous support, encouragement, and special considerations for my family during my Ph.D. journey. As a researcher, mentor, and entrepreneur, your working style has inspired me a lot.

I am grateful for having great Graduate Research Committee (GRC) Members: Prof. Stefan Decker, Dr. Paul Buitelaar, Dr. Conor Hayes, and Dr. Brian Davis, who have given inspiring discussions and advice on my research every year. The discussions I had with them in each GRC meeting reviewed my progress of each year critically, and made me shape my research directions for the next year efficiently. In addition, I would like to thank my thesis examiners Prof. Dr. Harith Alani and Prof. Dr. Dietrich Rebholz-Schuhmann who gave a lot of constructive feedback to improve the quality of this thesis.

I am grateful to my colleagues at Insight for the many inspiring conversations and exciting activities. Thanks to the unit members Ihab, Safina, Sebastian, Peiman, and Dr. Subhasis, and best wishes for your research and families.

I really appreciate all the mentors at Yanbian University of Science and Technology such as Byeongguk Ku, SeungHun Baek, Jooyeon Lee, and Yusin Park, who motivated me to study abroad and to share what you have learned in your life with other people. I am very thankful that having Prof. Wooju Kim and June S. Hong as my supervisors for my master's degree.

And most importantly, I would like to thank my wife Hua and my son Zhiyu who have lost many things in their lives for my research. Big thanks to Hua, my parents and brother who had given unconditional support and love in the past few years. It would not have been possible to make it without your support and love.

Finally, I would like to thank everyone who has supported me to make the thesis possible.

Contents

List of Figures

List of Tables

List of Abbreviations

BPR	**B**aysian **P**ersonalized **R**anking
CF	**C**oncept **F**requency
CDF	**C**umulative **D**istribution **F**unction
CNNs	**C**onvolutional **N**eural **N**etworks
CBRS	**C**ontent **B**ased **R**ecommender **S**ystem
CFRS	**C**ollaborative **F**iltering **R**ecommender **S**ystem
CoFM	**C**o-**F**actorization **M**achine
FTF	**F**ollowees' **T**erm **F**requency
FM	**F**actorization **M**achine
IDF	**I**nverse **D**ocument **F**requency
IP	**I**nterest **P**ropagation
KB	**K**nowledge **B**ase
KG	**K**nowledge **G**raph
LOD	**L**inked **O**pen **D**ata
LSTM	**L**ong **S**hort-**T**erm **M**emory network
LDSD	**L**inked **D**ata **S**emantic **D**istance
LODFM	**L**inked **O**pen **D**ata-enabled **F**actorization **M**achine
MRR	**M**ean **R**eciprocal **R**ank
nDCG	**N**ormalized **D**iscounted **C**umulative **G**ain
OSN	**O**nline **S**ocial **N**etwork
POI	**P**oint **O**f **I**nterest
RDF	**R**esource **D**escription **F**ramework
RNN	**R**ecurrent **N**eural **N**etwork
RS	**Rec**commender **Sys**tem
SGD	**S**tochastic **G**radient **D**escent
SVM	**S**upport **V**ector **M**achine
SA	**S**preading **A**ctivation
THT	**T**opic **H**ierarchy **T**ree
TF	**T**erm **F**requency
UGC	**U**ser **G**enerated **C**ontent
URL	**U**niform **R**esource **L**ocator
URI	**U**niform **R**esource **I**dentifier
UM	**U**ser **M**odeling
VSM	**V**ector **S**pace **M**odel

WSD	**W**ord **S**ense **D**isambiguation
WiBi	**W**ikipedia **Bi**taxonomy

Chapter 1

Introduction

1.1 Motivation

Online Social Networks (OSNs) such as Twitter[1] and Facebook[2] have been growing rapidly since they first emerged in the early 2000's, and are widely used in our daily lives. For example, Twitter and Facebook have 328 million and 2.01 billion monthly active users[3][4]. A recent survey also reveals that over 50% of users consume news in OSNs such as Twitter[5], which shows the popularity of these services.

On the one hand, the abundant information generated by users in OSNs creates new opportunities for inferring user interest profiles, which can be used for providing personalized recommendations to those users either on those OSNs or on third-party services allowing social login functionality[6] from the same OSNs. Social login is a technology which allows visitors to a website to log in using their OSN accounts rather than having to register a new one[7]. Figure 1.1 shows social login options along with a regular sign-in and registration option, as found in many applications or websites nowadays.

For example, a third-party application which provides news recommendations can utilize user interest profiles constructed from Twitter for personalized recommendations once a user has logged in using the social login functionality via his/her Twitter account. A recent survey showed that over 94% of 18-34 year olds have used social login via Twitter, Facebook, etc.[8], and another study from LoginRadius[9] showed that 94% of users use their OSN

[1] https://twitter.com/
[2] https://www.facebook.com/
[3] https://www.omnicoreagency.com/twitter-statistics/
[4] https://www.omnicoreagency.com/facebook-statistics/
[5] https://goo.gl/WsPrMS
[6] https://en.wikipedia.org/wiki/Social_login
[7] https://hbr.org/2011/10/social-login-offers-new-roi-fr
[8] http://www.gigya.com/blog/why-millennials-demand-social-login/
[9] https://www.loginradius.com/

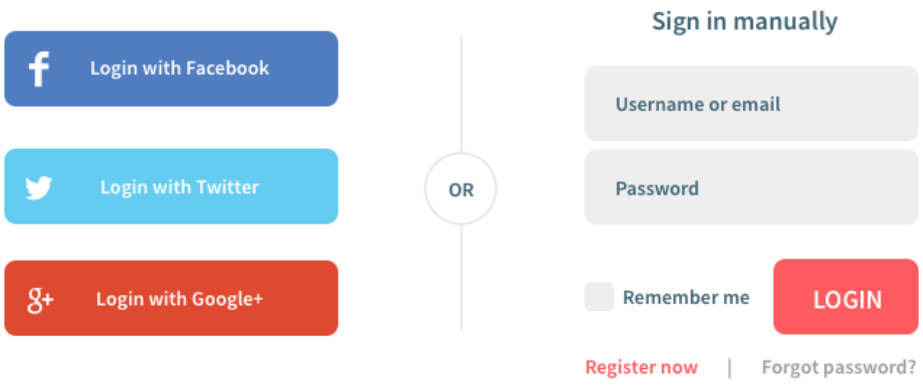


FIGURE 1.1: An example of social login page with a traditional login option via registration.

identities to log in websites based on the data collected from over 160,000 websites[10]. This indicates the importance of leveraging user profiles from OSNs for personalization with the permission of the users.

On the other hand, the huge volume of user-generated content causes an *information overload* problem for users who are trying to consume relevant information that they might be interested in. It has been reported that users follow 80 people on average on Twitter (Qu and Liu, 2011), which results in hundreds or even thousands of tweets being shown to each user every day.

Recommender systems, which suggest a few data points out of a large pool of data, play an important role in dealing with the information overload problem in OSNs as well as other domains such as e-commerce. Take LinkedIn[11] as an example: its product "people you may know" recommends only a few members out of a database of 300,000,000 members[12]. There are several key challenges for recommender systems in different scenarios.

1. A typical challenge is the *cold-start* problem (Schein et al., 2002), which denotes lacking explicit feedback from users. For example, it is difficult for a recommender system to provide recommendations without any explicit feedback from users, which is common for new users who have just started using the recommender system. Therefore, inferring user interest profiles from implicit feedback, e.g., user activities in OSNs, plays an important role in this type of *cold-start* problem.

2. Another *cold-start* scenario is when the system has only a limited number of explicit feedback instances from users, e.g., there exists

[10] https://blog.loginradius.com/2016/04/customer-identity-preference-trends-q1-2016/
[11] https://www.linkedin.com/
[12] https://www.forbes.com/sites/lutzfinger/2014/09/02/recommendation-engines-the-reason-why-we-love-big-data/#50b79da21077

only one liked item for each user in the initial state of a recommender system. In this case, it is useful to recommend similar items to the liked one of each user based on their similarities.

3. When there exists plenty of explicit feedback from users with respect to items, the key challenge for a recommender system is how to build a recommendation model based on the explicit feedback as training data.

Linked Open Data (LOD) (Bizer et al., 2009) indicates a new generation of technologies responsible for the evolution of the current Web from a Web of interlinked documents to a Web of interlinked data (Heath and Bizer, 2011). It provides a large amount of machine-consumable background knowledge in various domains, and is freely accessible on the Web. For example, cross-domain Knowledge Graphs (KGs) such as DBpedia (Auer et al., 2007), domain-specific data such as DrugBank[13], and more can be discovered at `http://lod-cloud.net/`. Therefore, LOD is a valuable source of information for providing background knowledge regarding user interests or items, which would be helpful for cold-start scenarios in recommender systems (Orlandi, 2014). Indeed, with the increasing number of open knowledge sources powered by LOD, there have been many novel studies leveraging semantic techniques that shift from a *keyword-based* to a *concept-based* representation of items and user profiles for recommender systems (Ricci et al., 2011).

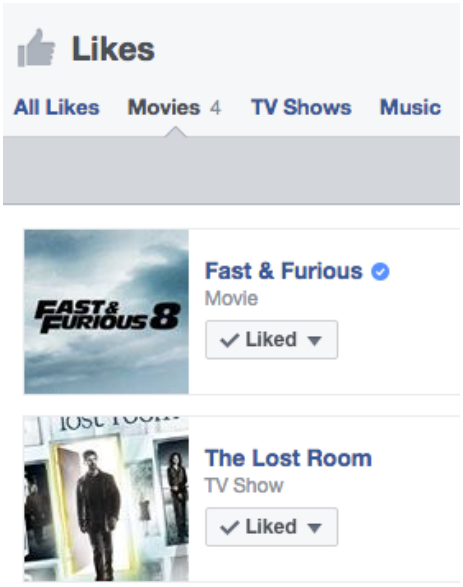
When there is plenty of explicit feedback from established users with respect to items such as movies or music artists, a common technique is using a collaborative filtering technique for recommender systems. A popular type of approach in collaborative filtering is *model-based* approaches, which factorizes users and items in high-dimensional vector spaces. Figure 1.2 shows an example of explicit feedback about movies from users on Facebook.

1.2 Objectives

In this thesis, we propose various *semantics-aware* approaches for the aforementioned three challenges. Semantics-aware approaches for recommender systems can be classified into two categories (Ricci et al., 2011, p. 119):

1. *Top-down* approaches that rely on the integration of external knowledge such as from KGs, and

[13] `https://old.datahub.io/dataset/fu-berlin-drugbank`



(A)

FIGURE 1.2: Example of explicit feedback from users about movies on Facebook in the context of recommending Facebook pages that a user might like.

2. *Bottom-up* approaches that exploit the so-called geometric metaphor of meaning to represent complex relations between users/items in high-dimensional vector spaces.

First, we explore *top-down* approaches with a focus on how we can infer and enhance user interest profiles in OSNs, leveraging background knowledge from KGs. For example, given a tweet like "My Top 3 #lastfm Artists: Eagles of Death Metal(14), The Black Keys(6) & The Wombats(6)" from a user, we can infer the user is interested in `dbr`[14]`:Indie_rock` in addition to `dbr:The_Wombats`, `dbr:The_Black_Keys`, and `Eagles_of_Death_Metal` as both `dbr:The_Wombats` and `dbr:The_Black_Keys` are pointing to `dbr:Indie_rock` via `dbo`[15]`:genre` in DBpedia. This propagation of user interests using background knowledge from KGs such as DBpedia can play a crucial role in inferring user interest profiles based on microblogs in OSNs such as tweets with short content.

In addition, we explore different dimensions of user modeling such as *user representation* strategies and *temporal dynamics* of user interests. In addition, we investigate the synergistic effect of considering multiple dimensions together for inferring user interests. The goal is to build qualified user interest profiles based on their activities in OSNs, which can be used for resolving the cold-start problem inside or outside of these OSNs.

[14]The prefix `dbr` denotes for http://dbpedia.org/resource/
[15]The prefix `dbo` denotes for http://dbpedia.org/ontology/

Secondly, we propose a semantic similarity/distance measure for measuring the semantic similarity between two items/entities in DBpedia based on their background knowledge. The similarity scores between entities can be used for recommending similar items for new users who lack of enough explicit feedback for collaborative filtering approaches.

Finally, we explore *bottom-up* approaches to learn latent semantic representations about users and items with the background knowledge of those items from KGs such as DBpedia for providing item recommendations when there is plenty of explicit feedback from users. To this end, we propose *LODFM*, which uses state-of-the-art factorization techniques such as factorization machines (Rendle, 2010) with LOD-enabled features. In addition, most previous studies including *LODFM* have not considered the *incompleteness* of KGs when exploring the background knowledge about items. In order to incorporate the incompleteness of KGs for item recommendations, we investigate transfer learning between the two tasks: (1) item recommendations, and (2) KG completion with respect to the domain of items.

1.3 Thesis Outline

This thesis is comprised of seven chapters. **Chapter 1** introduces the motivation and outline of the thesis with publication details for each chapter. **Chapter 2** provides some background and related work. The main contributions of this thesis are presented in **Chapters 3-6**. Figure 1.3 shows the overview of these chapters.

Chapters 3 and **4** mainly discuss user modeling strategies in OSNs for inferring user interest profiles based on *implicit feedback* from users such as their tweets or follow relationships in OSNs. In order to evaluate user interest profiles inferred by different user modeling strategies, we compare different profiles in the context of URL recommendations on Twitter where these profiles are used as an input to the URL recommendation system. URL represents a common "unit" of information on the Web, and has been used for evaluating different user modeling strategies in OSNs (Chen et al., 2010).

In **Chapters 5** and **6**, we investigate semantics-aware recommendation approaches based on *explicit feedback* from users. For example, users on Facebook like Facebook pages with respect to many domains such as movies (see Figure 1.2), musics, and books. **Chapters 5** and **6** focus on item recommendations in the music, movie, and book domains based on the items have been liked by users (explicit feedback). In each chapter, we first present the motivation and contributions which will be discussed in that chapter.

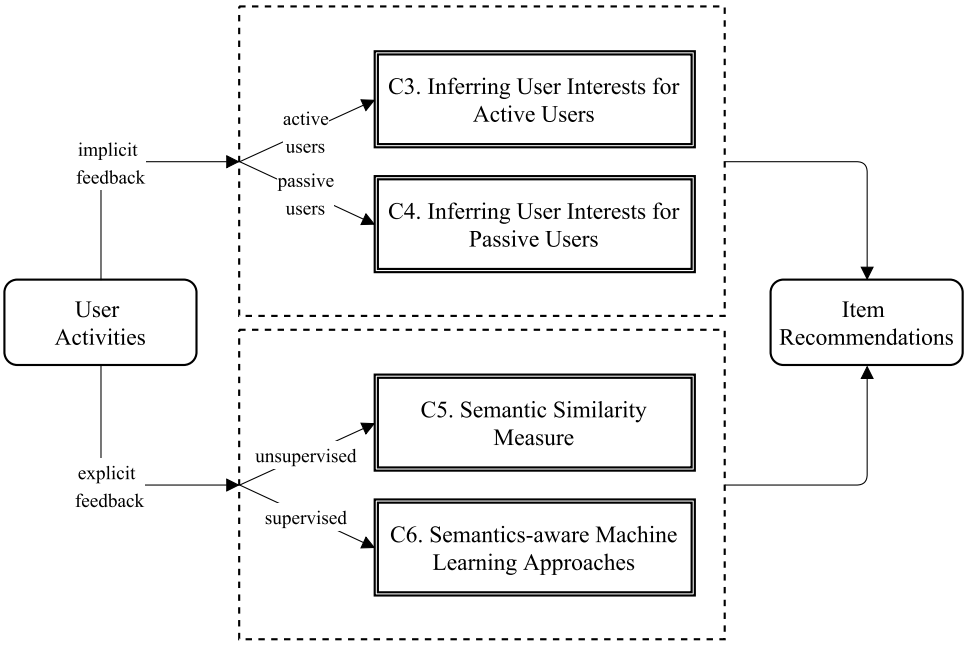


FIGURE 1.3: Overview of chapters corresponding to the main contributions.

Then we provide details of our conducted studies, and a summary of main findings based on these studies.

In **Chapter 2**, we provide an overview of related work on user modeling and recommender systems in OSNs. On top of that, we list the research gaps in previous studies which we aim to address within this thesis in Section 2.5.

We address the first research challenge presented in Section 1.1 in **Chapters 3-4**. In **Chapter 3**, we investigate various user modeling strategies across different dimensions for inferring user interest profiles, with the main focus on *active users* who consistently generate content on microblogging social networks. Section 3.2 describes how different user modeling strategies are evaluated in this thesis, and gives details of a Twitter dataset which we will use in many experiments throughout this thesis. Section 3.3 provides an overview of *entity*- and *category*-based user modeling strategies, and Section 3.4 introduces a weighting scheme for these strategies. In Section 3.5, we discuss interest propagation strategies using the DBpedia knowledge graph. Section 3.6 provides a comparative study of different approaches for incorporating the temporal dynamics of user interests, and Section 3.7 investigates representation strategies for user interest profiles. In Section 3.8, we provide a study of comprehensive user modeling strategies in order to investigate the synergistic effect of considering different user modeling dimensions altogether, and we then summarize the main findings in Section 3.9.

In **Chapter 4**, we investigate user modeling strategies for *passive users* who use OSNs for receiving information they need but who do not generate any content on OSNs. Section 4.2 proposes exploring the *biographies* of followees for inferring user interest profiles, and Section 4.3 investigates user modeling strategies which leverage the *list memberships* of followees for inferring user interests for passive users. In Section 4.4, we investigate the synergistic effect of combining two different views of followees based on their biographies and list memberships when inferring user interests for passive users. Finally, we summarize Chapter 4 in Section 4.5.

In **Chapter 5**, we introduce a semantic similarity measure, named *mLDSD*, for tackling the second research challenge mentioned in Section 1.1 when there is a limited number of explicit feedback instances from users. *mLDSD* measures the similarity between entities in a linked open dataset such as DBpedia, and those semantic similarity scores can be used for recommending similar items for an item that a user has liked in the past. In Section 5.2, we first introduce the Linked Data Semantic Distance (*LDSD*) measure (Passant, 2010b), and then present the components of *mLDSD*. Section 5.3 provides a preliminary evaluation of *mLDSD*. In Section 5.4, we present two refined Linked Data Semantic Distance measures for *mLDSD*, and evaluate them in Section 5.5 using a Facebook dataset in the context of a LOD-enabled recommender system (LODRecSys). In Section 5.6, we investigate the "Linked Data sparsity problem" in LOD-enabled recommender systems which are based on semantic similarity/distance measures, and Section 5.7 summarizes Chapter 5.

Regarding the third research challenge for building an accurate model based on users' explicit feedback, **Chapter 6** provides some semantics-aware machine learning approaches for recommending items in OSNs such as movies a user might like on Facebook. In Section 6.2, we propose *LODFM* which leverages factorization machines with lightweight LOD-enabled features from DBpedia for providing item recommendations for items such as movies. In Section 6.3, we propose a co-factorization approach for transfer learning between the two tasks: (1) item recommendations, and (2) knowledge graph completion in order to take into account the *incompleteness* of KGs.

Finally, **Chapter 7** summarizes our main findings and contributions by addressing the research challenges raised at the end of **Chapter 2**, and provides some possible directions for future work on top of the main findings in this thesis, particularly in the areas of user modeling and recommender systems in the domain of OSNs.

1.4 Overview of Datasets and Entity Recognition

To study and evaluate different user modeling strategies in Chapter 3 and Chapter 4, we use a Twitter dataset which we describe in Section 1.4.1. In addition, we discuss different entity recognition APIs for extracting DBpedia entities from tweets in Section 1.4.2. Finally, we provide the details of datasets used for LODRecSys in Section 1.4.3.

1.4.1 Twitter Dataset for Chapters 3-4

We built a Twitter dataset based on an about.me[16] dataset crawled in our previous work (Piao and Breslin, 2016a). About.me is a personal web hosting service which allows users to link multiple OSN identities such as Facebook, LinkedIn, and Twitter. The about.me dataset consists of 247,630 public profile pages retrieved from the service during December 2014.

We randomly selected 480 *active* users on Twitter, and further crawled all tweets from those active users via the Twitter API[17]. Due to the limit of the Twitter API, we are able to crawl up to the last 3,200 tweets posted by each user. The main details of the dataset are presented in Table 1.1. As we can see from the table, the 480 users posted 348,544 tweets in total in their timelines, and each user posted 726 tweets on average.

TABLE 1.1: Twitter dataset statistics.

# of users	480
total # of tweets	348,554
average time span of tweets per user (days)	471
average # of tweets per user	726
average # of tweets per user per day	7.2

This dataset is used for studying different user modeling strategies in Chapters 3 and 4. However, we will sample a subset of this dataset or crawl additional information which is missing in this dataset for different experiments. For the reader's convenience, we will describe the subset used in each experiment in Section 3.3.1, Section 3.4.1, Section 4.2.3, and Section 4.3.3, respectively.

[16] https://about.me/
[17] https://dev.twitter.com/rest/public

To evaluate inferred user interest profiles based on different user modeling strategies, we use these profiles as an input to an URL recommender system on Twitter (other evaluation strategies for user modeling can be found in Section 2.3.4). For a target user, the *ground truth* URLs (the positive set) are the ones shared by this user in his/her tweets. In addition, those URLs shared by other users in the dataset but not by the target user are used as the negative set of URLs. Therefore, a user interest profile should be able to rank the URLs in the positive set higher than those in the negative set.

1.4.2 Entity Recognition APIs

Here we investigate different entity recognition APIs for extracting DBpedia entities from tweets. Those extracted DBpedia entities play an important role in retrieving user interests in semantics-aware user modeling strategies introduced in this thesis, which can be used for propagating user interests by leveraging background knowledge in DBpedia.

Entity recognition in tweets is a challenging task due to the informal nature of and ungrammatical language in tweets. Since our focus in this thesis is on user modeling and not on entity recognition, we use an existing solution for entity recognition (as does related literature on user modeling).

Different NLP APIs have been used for DBpedia/Wikipedia entity recognition in the literature. For example, Kapanipathi et al., 2014 used the Zemanta API (which is no longer available at the time of writing.) after comparing it to other APIs such as DBpedia Spotlight[18], while Zarrinkalam and Kahani, 2015 used tag.me[19]. To better investigate the performance of different APIs, we used a Twitter dataset from Locke, 2009 which contains 1,603 annotated tweets in total where 1,233 of them contain Wikipedia entities. We tested three different NLP APIs: Aylien API[20], tag.me and Alchemy API[21], which all provide the functionality for extracting entities from a given text and representing these with corresponding DBpedia/Wikipedia URIs. A comparative performance of these three APIs on entity (with DBpedia URI) recognition is displayed in Table 1.2.

We opted to use the Aylien API since (1) it extracts DBpedia entities identified in a text, and gives their corresponding URIs, (2) it has relatively superior performance to the other APIs as shown in Table 1.2, and (3) it provides 6,900 calls per day, provided on request for research purposes.

[18] http://spotlight.dbpedia.org/rest/annotate, the web service was not accessible at the time of writing this thesis.
[19] https://tagme.d4science.org/tagme/
[20] https://aylien.com/
[21] http://www.alchemyapi.com/

TABLE 1.2: Evaluation of several NLP APIs for DBpedia/Wikipedia entity recognition.

API	Precision	Recall	F-measure
Aylien	0.27	0.26	0.26
Alchemy	0.21	0.17	0.19
tag.me	0.12	0.15	0.14

The Aylien API is used for extracting DBpedia entities identified in *tweets* in Chapter 3 as well as those entities identified in the *biographies* of a user's followees in Chapter 4. In addition, this API is also used for extracting DBpedia entities for URLs in Chapters 3 and 4. The Aylien API provides the functionality of DBpedia entities for a given URL based on its content.

1.4.3 Datasets for Chapters 5-6

As discussed in Section 1.3, Chapters 5-6 deal with the scenarios when we have explicit feedback from users, e.g., which Facebook pages should we recommend based on the pages liked by users with respect to musical artists. The list of datasets used in Chapters 5-6 is presented in Figure 1.3. The items available in these datasets have been mapped to their corresponding DBpedia URIs.

The mapped Facebook dataset[22] consists of 52,072 users and 6,375 items where each user has liked 21 items on average. This dataset was collected from Facebook profiles about personal preferences ("likes") with respect to musical artists. This dataset is used in the experiments in Section 5.5.1 and Section 6.2.3.

The mapped Movielens dataset consists of users and their ratings about movie items, and we consider ratings higher than 3 as positive feedback in the same way as Noia et al., 2016. As we can see from Table 1.3, this dataset consists of 3,997 users and 3,082 items where each user has rated 206 movies. This dataset is used in the experiments in Section 6.2.3 and Section 6.3.2.

The mapped DBbook dataset[23] consists of users and their binary feedback (1 for likes, and 0 otherwise) with respect to book items. DBbook dataset consists of 6,181 users and 6,733 items where each user has rated 12 books. This dataset is used in the experiments in Section 6.3.2.

[22] https://2015.eswc-conferences.org/important-dates/call-RecSys.html
[23] http://challenges.2014.eswc-conferences.org

TABLE 1.3: Statistics of Facebook, Movielens and DBbook datasets.

	Facebook	**Movielens**	**DBbook**
# of users	52,072	3,997	6,181
# of items	6,375	3,082	6,733
# of ratings	1,093,512	827,042	72,372
avg. # of ratings	21	206	12
sparsity	99.67%	93.27%	99.38%
% of positive ratings	100%	56%	45.85%

We evaluate semantics-aware recommendation approaches such as the semantic similarity measure proposed in Chapter 5 and machine learning approaches proposed in Chapter 6 in terms of item recommendation using aforementioned datasets in different domains. For a target user, the *ground truth* items (the positive set) are the ones with positive feedback from this user. In addition, those items in the dataset but not rated by the target user are used as the negative set of items. Therefore, the proposed recommendation approaches should be able to rank the items in the positive set higher than those in the negative set.

1.5 Origin of Chapters

In this section, we introduce the publications associated with each chapter for the five core chapters (**Chapters 2-6**). Every publication has been published in peer-reviewed conferences related to the research topics in this thesis. Some passages in each chapter have been quoted verbatim from the these sources.

Chapter 2

- **G. Piao**. Towards Comprehensive User Modeling on the Social Web for Personalized Recommendations. The Doctoral Consortium at the 24th Conference on User Modeling, Adaptation and Personalization (**UMAP 2016**), Halifax, Canada, 2016 - (Piao, 2016b)
- **G. Piao**. Exploiting the Semantic Similarity of Interests in A Semantic Interest Graph for Social Recommendations: Student Research Abstract. The Student Research Competition at the 31st Annual ACM

Symposium on Applied Computing (**SAC 2016**), [**Finalist**], Pisa, Italy, 2016 - (Piao, 2016a)

- **G. Piao, J. G. Breslin**. Inferring User Interests in Microblogging Social Networks: A Survey (accepted in User Modeling and User-Adapted Interaction) - (Piao and Breslin, 2018e)

Chapter 3

- **G. Piao**, J. G. Breslin. Analyzing Aggregated Semantics-enabled User Modeling on Google+ and Twitter for Personalized Link Recommendations. In Proceedings of the 24th Conference on User Modeling, Adaptation and Personalization (**UMAP 2016**), Halifax, Canada, 2016 - (Piao and Breslin, 2016a)

- **G. Piao**, J. G. Breslin. User Modeling on Twitter with WordNet Synsets and DBpedia Concepts for Personalized Recommendations. In Proceedings of the 25th ACM International Conference on Information and Knowledge Management (**CIKM 2016**), Indianapolis, USA, 2016 - (Piao and Breslin, 2016d)

- **G. Piao**, J. G. Breslin. Exploring Dynamics and Semantics of User Interests for User Modeling on Twitter for Link Recommendations. In Proceedings of the 12th International Conference on Semantic Systems (**SEMANTiCS 2016**), [**Best Paper Award**], Leipzig, Germany, 2016 - (Piao and Breslin, 2016b)

- **G. Piao**, J. G. Breslin. Interest Representation, Enrichment, Dynamics, and Propagation: A Study of the Synergetic Effect of Different User Modeling Dimensions for Personalized Recommendations on Twitter. In Proceedings of the 20th International Conference on Knowledge Engineering and Knowledge Management (**EKAW 2016**), Bologna, Italy, 2016 - (Piao and Breslin, 2016c)

Chapter 4

- **G. Piao**, J. G. Breslin. Inferring User Interests for Passive Users on Twitter by Leveraging Followee Biographies. In Proceedings of the 39th European Conference on Information Retrieval (**ECIR 2017**), Aberdeen, UK, 2017 - (Piao and Breslin, 2017b)

- **G. Piao**, J. G. Breslin. Leveraging Followee ListMemberships for Inferring User Interests for Passive Users on Twitter. In Proceedings of the 28th ACM Conference on Hypertext and Social Media (**HT 2017**), Prague, Czech, 2017 - (Piao and Breslin, 2017c)

Chapter 5

- **G. Piao**, J. G. Breslin. Computing the Semantic Similarity of Resources in DBpedia for Recommendation Purposes. In Proceedings of the 5th Joint International Semantic Technology Conference (**JIST 2015**), [**Best Paper Candidate**], Yichang, China, 2015 - (Piao et al., 2015)
- **G. Piao**, J. G. Breslin. Measuring Semantic Distance for Linked Open Data-enabled Recommender Systems. In Proceedings of the 31st ACM/SIGAPP Symposium on Applied Computing (**SAC 2016**), Pisa, Italy, 2016 - (Piao and Breslin, 2016f)

Chapter 6

- **G. Piao**, J. G. Breslin. Factorization Machines Leveraging Lightweight Linked Open Data-enabled Features for Top-N Recommendations. In Proceedings of the 18th International Conference on Web Information Systems Engineering (**WISE 2017**), Moscow, Russia, 2017 - (Piao and Breslin, 2017a)
- **G. Piao**, J. G. Breslin. Transfer Learning for Item Recommendations and Knowledge Graph Completion in Item Related Domains via a Co-Factorization Model. In Proceedings of the 15th Extended Semantic Web Conference (**ESWC 2018**), Crete, Greece, 2018 - (Piao and Breslin, 2018f)

Chapter 2

Background

In this chapter, we introduce some background with respect to semantics-aware user modeling and recommender systems (Section 2.3-2.4) in the context of online social networks. We start by giving a brief overview of online social networks and recommender systems (Section 2.1-2.2). Next, we will present related work on semantics-aware user modeling and recommender systems with a focus on the OSN domain. Finally, we summarize the research challenges that we aim to address in Section 2.5.

2.1 Online Social Networks

Online Social Networks (OSNs, also social networking sites or services, SNS or social media) are online platforms that people use to build their social networks or social relations with other people who share similar personal or career interests, activities, backgrounds or real-life connections[1]. Although it is challenging to give a unanimous definition of OSNs due to the variety of these services, there are some commonalities among them (Obar and Wildman, 2015):

- Social media services are (currently) Web 2.0 Internet-based applications.
- User-Generated Content (UGC) is the lifeblood of social media.
- Individuals and groups create user-specific profiles for a site or app designed and maintained by a social media service.
- Social media services facilitate the development of social networks online by connecting a profile with those of other individuals and/or groups.

[1] https://en.wikipedia.org/wiki/Social_networking_service

On the one hand, online social networks such as Facebook or Twitter have been embedded into our daily lives. Facebook, launched in 2004, has 2.01 billion monthly active users all over the world; Twitter, launched in 2006, currently has 328 million monthly active users posting over 500 million tweets every day[2]. Figure 2.1 shows the increasing number of social media users worldwide from 2010, which is provided by statista[3]. The identities and content generated by users in these OSNs provides a great opportunity for various valuable applications, such as event detection from Twitter streams for early warning (Sakaki et al., 2010), discovery of fresh Web sites (Dong et al., 2010), analyzing crisis information (Burel et al., 2017), and inferring user interests for personalized recommendations. Online social networks have also become important platforms for users to consume different types of information such as news or medical information. One in three Web users seeks medical information in OSNs, and over 50% of users consume news in OSNs[4] (Sheth and Kapanipathi, 2016).

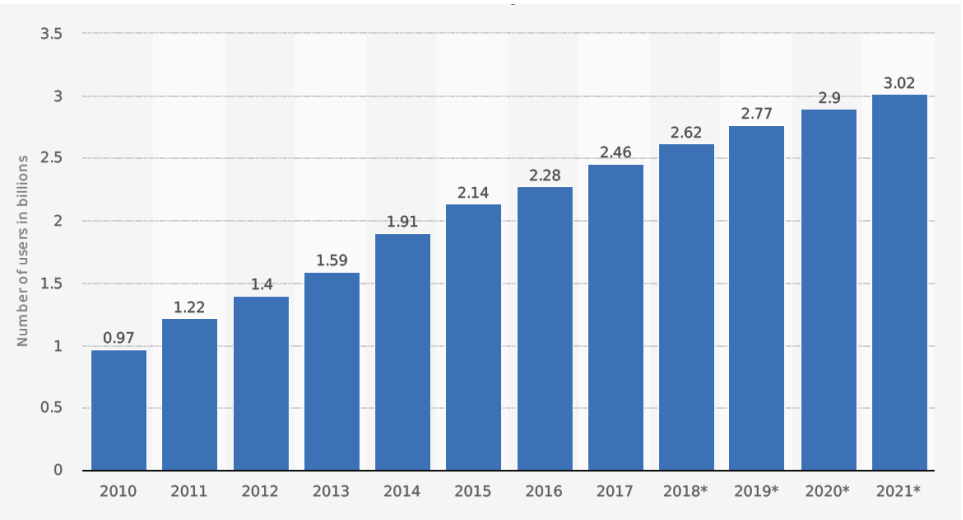


FIGURE 2.1: Number of social media users worldwide from 2010 to 2021 (in billions).

On the other hand, the tremendous popularity of these services has led to an increase of user-generated content, which in turn has caused information overload for users, and made it difficult for them to consume information that they might be interested in. For example, users are often overwhelmed by the large number of tweets from their followees on Twitter. Therefore, recommending content matching users' interests is crucial for them when consuming information on these services.

[2] https://www.omnicoreagency.com/twitter-statistics/

[3] https://goo.gl/PdZthX, statista is a statistics, market research and business intelligence portal.

[4] https://goo.gl/t7hmPQ

2.2 Recommender Systems

A recommender system or a recommendation system (sometimes replacing "system" with a synonym such as platform or engine) is a subclass of the information filtering system that aims at predicting the "rating" or "preference" of an item for a user, and recommends items according to their "ratings" or "preference" scores[5]. In other words, it is trained based on explicit feedback (i.e., item ratings or preference history from users) for predicting the "rating" or "preference" of an item, and then retrieves items that users would like in the future based on the "ratings" or "preference" scores of those items.

Recommender systems (RSs) have been used in a variety of applications such as product recommendations in Amazon[6], and post or friend recommendations in OSNs such as Twitter. There are three major approaches for recommender systems: (1) content-based recommender systems (CBRSs), (2) collaborative filtering recommender systems (CFRSs), and (3) hybrid recommender systems.

A CBRS recommends items which are similar to the ones that a user has liked in the past. This type of RS aims to match the attributes of a user and an item, where these attributes can be extracted from the content (e.g., descriptions) of items. To this end, a user profile is constructed based on the content of the items that the user has liked in the past. For example, a movie genre might be an important attribute with respect to movie recommendations, and a user profile might contain the movie genres that the user has liked. Classic CBRSs extract *keywords* from items' descriptions and use these keywords as attributes.

More recently, semantics-aware techniques have arisen to cope with the main problems of classical keyword-based approaches (Ricci et al., 2011, p. 12). As we described in Section 1.1, there are two main categories of semantics-aware techniques. The first one is *top-down* approaches, which integrate external knowledge sources such as an encyclopedia (e.g., Wikipedia[7]) or an open knowledge graph (e.g., DBpedia). The second category is *bottom-up* approaches, which learn the representation of words based on their usage in a large corpus of textual documents.

In contrast to CBRSs, a CFRS makes recommendations based on the assumption that if a user A has the same preference as a user B on an item i, A tends to have the same preference as B on a different item j than a

[5] https://en.wikipedia.org/wiki/Recommender_system
[6] https://www.amazon.com/
[7] https://www.wikipedia.org/

randomly chosen item. There are two types of approaches in CFRSs: (1) memory-based and (2) model-based approaches. *Memory-based* approaches leverage user preference data to compute the similarities between users or items, and provide recommendations based on these similarity scores. In contrast, *model-based* approaches predict the preference score of a candidate item based on building a model using different data mining and machine learning algorithms, e.g., latent factor models such as matrix factorization.

Hybrid recommender systems combine several techniques used in different RSs such as CBRSs or CFRSs for providing recommendations in order to overcome the disadvantages in a RS with the advantages from another RS. For example, a common cold-start problem in CFRSs is predicting the preference scores of *new items* as there is no preference data from other users for these items. In contrast, CBRSs do not suffer from this new-item problem as they rely on features of items for providing recommendations. More details on various approaches for combining several techniques from different RSs can be found in Brusilovsky et al., 2007, p. 377-408.

Another cold-start scenario occurs when recommending items to *new users* who have not provided any feedback about items yet, which is common for the initial stage of many websites or mobile applications. Inferring user interest profiles from user-generated content in OSNs can provide much information about users, and this information can resolve the new-user problem for those websites or applications using social login functionality. Social login is a single sign-on functionality that allows users to use their existing OSN identities for signing in to a third-party website/application instead of creating a new account for that website/application[8]. According to a Web Hosting Buzz[9] survey, 86% of users reported that creating new accounts on different websites bothers them, and some of them responded that they would choose to leave a website because of it. 77% of respondents of the survey said that "Social login is a good solution that should be in any site". A study from Gigya in 2015 revealed the continued high growth use of social login functionality, and shows that 88% of US consumers have logged in to a website or application using their social network identities, which is an increase of 11% compared to a study from the previous year[10].

The popularity of OSNs and the increased adoption of social login functionality create a new opportunity for those third-party websites or applications to provide personalized services based on user interests inferred from their OSN actives. Therefore, user modeling in OSNs for inferring user interests is important for those third-party applications as well as OSNs in order to

[8] https://auth0.com/learn/social-login/
[9] https://www.webhostingbuzz.com/
[10] https://goo.gl/BBGtdg

provide good recommendations in a cold start situation. We define user modeling in this thesis in the following section before reviewing related work on user modeling in OSNs.

2.2.1 User Modeling

A *user model* profiles the user with respect to his/her preferences and needs, and plays an important role in recommender systems. In a certain sense, a recommender system can be viewed as a tool that generates recommendations by building and exploring user models (Berkovsky et al., 2008; Berkovsky et al., 2009; Ricci et al., 2011).

Recommender systems can construct a user model/profile based on either *explicit* or *implicit* feedback. For example, in a tweet recommender system which aims at recommending tweets that could be retweeted by a user, the retweet histories of users can be used as their explicit feedback for constructing user models. For cold-start users that do not have any retweet history, we can still infer their interest profiles based on implicit feedback from them such as their tweets or social networks. As we already mentioned, user modeling in OSNs is not only important for providing recommendations in OSNs themselves but also plays an important role in making recommendations in third-party applications connected to these OSNs, e.g., via the social login functionality.

Rich, 1979 along with Cohen and Perrault, 1979 and Perrault et al., 1978, where the terms *user model* and *user modeling* can be traced back to, provide three major dimensions for classifying user models (Rich, 1979):

- Are they models of a canonical user or are they models of individual users?
- Are they constructed explicitly by the user themselves or are they abstracted by the system on the basis of the user's behavior?
- Do they contain short-term or long-term information?

Based on these three dimensions, we define *user model* in this thesis as below.

Definition 2.2.1 (User Model)**.** A *user model* is a representation of user interests about an individual user constructed either explicitly or implicitly based on long-term or short-term knowledge, and the process of obtaining the user model is called *user modeling*.

The terms "user model" and "user (interest) profiles" are used interchangeably throughout this thesis.

2.3 Inferring User Interest Profiles in Microblogging Social Networks

Here we provide background related to the first research challenge described in Section 1.1. In this thesis, we focus on *microblogging* social networks such as Twitter or Facebook. Although the character limit for a Facebook post is more than 60k, the average length of posts generated by users is smaller than 140 characters[11]. User modeling approaches for other OSNs such as Delicious[12] and Flickr[13] which are mainly based on *folksonomies* (folks taxonomies) (e.g., Hung et al., 2008; Szomszor et al., 2008; Abel, 2011; Cantador et al., 2008; Mezghani et al., 2012; Carmagnola et al., 2008, to name a few) are out of the scope of this thesis .

Although there are many choices of microblogging OSNs for investigating user modeling strategies, Twitter has been widely used in the literature due to its popularity and the higher degree of openness (in terms of data access). Table 2.1 provides a summary of OSNs used for inferring user interest profiles in previous studies. Other OSNs such as Facebook or LinkedIn requires the permissions of users to access their data. Therefore, users have to be recruited for conducting an experiment, which results in less studies using these OSNs.

Given the definition of a user model (Definition 2.2.1), Figure 2.2 presents an overview of the modified user profile-based personalization process from Abdel-Hafez and Xu, 2013 and Gauch et al., 2007a, which consists of three main phases. The first step is collecting data which will be used for inferring user interests. Subsequently, user interest profiles are constructed based on the data collected. We use *primitive interests* (Kapanipathi et al., 2014) to denote the interests directly extracted from the collected data. Those primitive interests can be used as the final output of a profile constructor or be further enhanced, e.g., based on background knowledge from Knowledge Bases (KBs) such as Wikipedia. We use *propagated interests* to denote the interests propagated by exploring the background knowledge based on the extracted primitive interests. The output of the profile constructor is user interest profiles represented based on a predefined representation of interest

[11] https://web.archive.org/web/20151204114826/https://www.quintly.com/blog/2013/12/short-posts-on-facebook-twitter-google-more-interactions

[12] https://del.icio.us/

[13] https://www.flickr.com/

TABLE 2.1: Online Social Networks used for previous studies.

OSNs	Examples
Twitter	Chen et al., 2010, Lu et al., 2012, Kapanipathi et al., 2014, Kapanipathi et al., 2011, Weng et al., 2010, Besel et al., 2016a; Besel et al., 2016b, Abel et al., 2011b Abel et al., 2011c; Abel et al., 2012; Abel et al., 2011a, Abel et al., 2013b, Siehndel and Kawase, 2012, Michelson and Macskassy, 2010, Bhattacharya et al., 2014, Orlandi et al., 2012, Hannon et al., 2012, Budak et al., 2014, Faralli et al., 2015b; Faralli et al., 2017, Zarrinkalam and Kahani, 2015; Zarrinkalam et al., 2016, Narducci et al., 2013, Xu et al., 2011, Jiang and Sha, 2015, Garcia Esparza et al., 2013, Gao et al., 2011, Nishioka and Scherp, 2016; Nishioka et al., 2015, Vu and Perez, 2013, Phelan et al., 2009, Peñas et al., 2013, Sang et al., 2015, Karatay and Karagoz, 2015, Kanta et al., 2012, O'Banion et al., 2012, Lim and Datta, 2013, Große-Bölting et al., 2015
Facebook	Kang and Lee, 2016, Orlandi et al., 2012, Kapanipathi et al., 2011, Narducci et al., 2013, Bhargava et al., 2015, Ahn et al., 2012
Google+[14]	Piao and Breslin, 2016a
LinkedIn	Kapanipathi et al., 2011

profiles, e.g., word-based user interest profiles. Finally, the constructed user profiles are evaluated, and can be used in specific applications such as recommender systems for personalized recommendations.

In the following, we discuss four dimensions of the user modeling process: (1) *data collection*, (2) *representation* of user interest profiles, (3) *profile construction and enhancement*, and (4) *evaluation* of the constructed user profiles.

[14] https://plus.google.com/

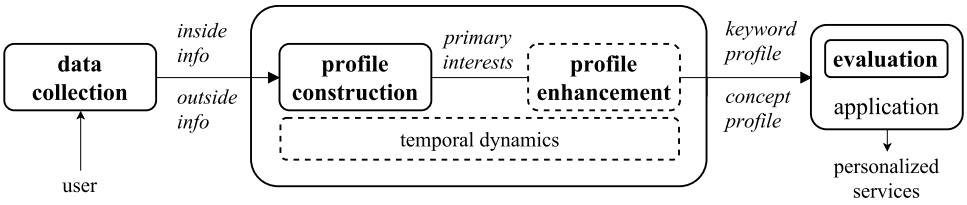


FIGURE 2.2: Overview of user profile-based personalization process.

2.3.1 Data Collection

What information is used for user modeling is important as it might directly affect later stages such as the representation and construction of user interest profiles, and the quality of final profiles. In the literature, the source of data collected for inferring user interest profiles is either from (1) *inside* of the target microblogging platform (where the target users come from) or (2) *outside* of the platform.

Using information inside the platform

For a target platform, various types of information from *users* and their *social networks* have been explored for inferring user interest profiles. Either by collecting data from a *target user* or the *social networks* of that user, there are three types of information sources for inferring user interest profiles:

- activities (e.g., tweets);
- profile information (e.g., biographies, account names);
- communities or groups (e.g., list memberships in Twitter).

A straightforward way of inferring user interests for a target user is leveraging information about the user. For instance, this could be analyzing data from the user's posts or profile, or topical groups such as *list memberships* in Twitter which the user belongs to. In contrast, for a *passive* user who has little activity but who keeps following other users to receive information, data from the user's social networks can be useful for indirectly inferring his/her interests. For instance, we can assume that a user is interested in `Microsoft` if a user is following the Twitter account *@Microsoft*, or is following an account with a biography such as "updates about Microsoft products". Also, user interests can be inferred indirectly by aggregating and analyzing posts or topical groups of a user's followees.

The posts generated by users are the most common source of information which has been used for inferring user interests. Take Twitter as an example, the tweets or retweets of users provide a great amount of data that might implicitly indicate what kinds of topics a user might be interested in. Therefore, using the post streams of target users to infer their interest profiles has been widely studied in the literature regardless of the different manners through which user interests are represented. For instance, Kapanipathi et al., 2014 extracted Wikipedia entities from the tweet streams of users while Chen et al., 2010 extracted keywords from them.

Inferring user interests based on users' posts requires users to be active, i.e., continuously generating content. On the one hand, there is an increasing number of users leveraging OSNs to seek information they need, e.g., one in three Web users look for medical information, and over half of surveyed users consume news in OSNs[15] (Sheth and Kapanipathi, 2016). On the other hand, there is also a rise of *passive users* in OSNs, e.g., two out of five Facebook users only browse information without interaction with the platform[16] (Besel et al., 2016a). Therefore, it is also important to infer user interest profiles for those *passive users* who consume information in OSNs without generating any content.

With the special characteristics of social networks, information from social networks such as tweets from followees or followers and posts from Facebook friends can be utilized for inferring user interests for *passive users* as well as *active users*. For instance, Chen et al., 2010 and Budak et al., 2014 explored the tweets of target users and their followees to infer user interests. Although using posts generated by users is of great potential for mining user interests, it also faces some challenges due to the short and noisy nature of microblogs. Some studies (Besel et al., 2016b; Faralli et al., 2015b; Faralli et al., 2017; Besel et al., 2016a) pointed out that exploring posts for inferring user interests is computationally ineffective and unstable due to the changing interests of users.

Instead of analyzing posts to infer user interests, these studies proposed using *followeeship* information of users, which can infer more stable user interest profiles as the relationships of common users tend to be stable (Myers and Leskovec, 2014). In this line of work, *topical followees* that can be mapped to Wikipedia entities often need to be identified, e.g., identifying the followee account *@messi10stats* on Twitter as `wiki`[17]`:Lionel_Messi`. Lim and Datta, 2013 also identified topical followees first, and then classified these followees into 15 predefined interest categories based on their "occupation"

[15] http://bit.ly/pewsnsnews
[16] http://www.corporate-eye.com/main/facebooks-growing-problem-passive-users/
[17] The prefix `wiki` denotes https://en.wikipedia.org/wiki/

fields and abstracts in Wikipedia. One of the problems in this approach is that only a small portion of users' followees are topical ones. For example, the authors from Faralli et al., 2015b showed that, on average, only 12.7% of followees of users in their datasets can be linked to Wikipedia entities. Similarly, in the Twitter dataset which we use for the experiment in Section 4.2.3, we observe that only 10% of followees' accounts can be mapped to Wikipedia entities.

List membership, which is a kind of "tagging" feature on Twitter, has been explored as well. A list membership is a topical list/group which can be generated by any user on Twitter, and the creator of the list can freely add other users to that topical list. For instance, a user *@Bob* might create a topical list named "Java" and add his followees or other users who have been frequently tweeting about news on this topic. Therefore, if a user *@Alice* is following users who have been added into many topical lists related to the topic `Java`, it might suggest that *@Alice* is also interested in this topic. Kim et al., 2010 studied the usage of Twitter lists and confirmed that lists can serve as good groupings of Twitter users with respect to their characteristics based on a user study. Based on the study, the authors also suggested that the Twitter list can be a valuable information source in many application domains including recommender systems. In this regard, several studies have exploited list memberships of followees to infer user interest profiles (Hannon et al., 2012; Bhattacharya et al., 2014).

User interests might follow global trends in some trends-aware applications such as news recommendations. To investigate it, Gao et al., 2011 proposed interweaving global trends and personal user interests for user modeling. In addition to leveraging the tweets of a target user for constructing user interest profiles, the authors constructed a trend profile based on all tweets in the dataset in a certain time period. Afterwards, the final user interest profile was built by combining the two profiles. The results showed that combined user interest profiles can improve the performance of news recommendations while the first user profile based on personal tweets plays a more significant role in the combination.

Using information outside the platform

The ideal length of a post on any OSN ranges between 60 to 140 characters for better user engagement[18]. Therefore, analyzing microblogging services such as Twitter is challenging due to their nature of generating short and noisy texts. Better understanding those short messages plays a key role in user modeling in microblogging services. To this end, previous

[18] `https://goo.gl/j97H1R`

studies have investigated leveraging external sources such as the content of embedded links/URLs in a tweet, in order to enrich the short text for a better understanding of it.

Haewoon et al., 2010 showed that most of the topics on Twitter are about news which could also be found in mainstream news sites. In this regard, some researchers have proposed linking microblogs to news articles and exploring the content of news articles in order to understand short texts in microblogging services. For instance, Abel et al., 2011b; Abel et al., 2011c; Abel et al., 2013b proposed linking tweets to news articles and extract the *primitive interests* of users based on tweets as well as the content of related news articles. Several strategies were proposed in Abel et al., 2011c, which were developed later on as a Twitter-based User Modeling Service (TUMS) (Tao et al., 2012). However, this type of approaches requires the maintenance of up-to-date news streams from mainstream news providers such as CNN[19] in order to link tweets to news articles. Instead, Abel et al., 2011a leveraged the content of the embedded URLs in tweets, and Hannon et al., 2012 used a third-party service Listorious[20], which is a service providing annotated tags of list memberships on Twitter, for inferring user interest profiles. Given a target user u, the authors construct u's interest profile based on the tags of list memberships with respect to the user.

With the popularity of different OSNs, users nowadays tend to have multiple OSN accounts across various platforms (Liu et al., 2013). In this context, some of the previous studies have investigated exploiting user interest profiles from other social networking platforms for cross-system user modeling. For instance, Orlandi et al., 2012 and Kapanipathi et al., 2011 presented user modeling applications that can aggregate different user interest profiles from various OSNs. However, the evaluation of aggregated user interest profiles has not been provided. Abel et al., 2012 investigated cross-system user modeling with respect to Points Of Interest (POI), and showed that the aggregation of Twitter and Flickr user data yields the best performance in terms of POI recommendations compared to modeling users separately based on a single platform. The result is in line with another study by them which aggregated user interest profiles on social tagging systems such as Delicious[21], StumbleUpon[22] (Abel et al., 2013a). Similar observations can be found in Piao and Breslin, 2016a, which investigated the aggregated user interest profiles from microblogging OSNs such as Twitter and Google+. Different from Abel et al., 2012 which aggregated different user interest profiles from different OSNs with the same weights, Piao and Breslin, 2016a

[19] http://edition.cnn.com/
[20] http://listorious.com, not available at the time of writing.
[21] https://www.delicious.com
[22] https://www.stumbleupon.com

showed that giving a higher weight to the target platform (with the aim of providing personalized services) is needed in order to provide the best performance in the context of URL recommendations.

2.3.2 Representation of User Interest Profiles

Here we provide an overview of how user interest profiles have been represented in the different approaches. The overview of user profile representation is carried out based on two criteria: (1) *a unit for representing user interests*, and (2) *polyrepresentation of user interest profiles*.

Unit for representing user interests

A *user interest unit* denotes the unit for representing user interests. For example, a single *word* is the interest unit for user interest profiles being represented as word vectors, and a *topic* in topic modeling approaches is the unit for those profiles being represented as topic vectors. In Gauch et al., 2007b, the authors defined three types of user representations for personalized information access:

- keyword profiles;
- concept profiles;
- semantic network profiles.

Keyword profiles. In keyword-based representation of user interest profiles, each *keyword* or a *group of keywords* can be used for representing a topic of interest. This approach was predominant in every adaptive information retrieval and filtering system and is still popular in these areas (Brusilovsky et al., 2007). When using each keyword for representing user interests, the importance of each word with respect to users can be measured using a defined weighting scheme such as TF·IDF (Term Frequency · Inverse Document Frequency) from information retrieval (Salton and McGill, 1986). In the case of using groups of keywords for representing user interests, the user interest profiles can be represented as a probability distribution over some topics, and each topic is represented as a probability distribution over a number of words. The topics can be distilled using topic modeling approaches such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003), which is an unsupervised machine learning method to learn topics from a large set of documents. One of the drawbacks of keyword-based user profiles is *polysemy*, i.e., a word may have multiple meanings which cannot be distinguished by using keyword-based representation.

Similar to other adaptive information retrieval and filtering systems, representing user interests using *keywords* or *groups of keywords* is popular in OSNs as well despite its simplicity. For instance, Chen et al., 2010 and Bhattacharya et al., 2014 represented user interest profiles by using vectors of weighted keywords extracted from users' tweets and the descriptions of list memberships of them, respectively. Another special type of keyword is *tags* (including *hashtags*). Hannon et al., 2012 pointed out that the user profiles constructed using keywords from microblogs can be large but noisy due to the challenges of understanding microblogging messages (Liao et al., 2012). As an alternative, instead of extracting keywords from the microblogs of users, Abel et al., 2011b; Abel et al., 2011a and Hannon et al., 2012 leveraged keywords from tags or hashtags for representing user interests. They suggest that keywords from tags/hashtags might be more informative and categorical in nature compared to the words mined from the short texts of microblogs.

Topics distilled from topic modeling approaches such as LDA are also popular for representing user interest profiles. A topic has associated words with their probabilities with respect to the topic. For example, an IT-related topic can have some top associated words such as "google, twitter, apple, web". Weng et al., 2010 used LDA to distill 50 topics and represented each user as a probability distribution over these topics. In Abel et al., 2011c; Abel et al., 2011b; Abel et al., 2013b, the authors also used topics for representing user interests where those topics were extracted by ready-to-use NLP (Natural Language Processing) APIs such as OpenCalais[23]. These keyword-based approaches lack semantic information and cannot capture relationships among these words, and the assumption of topic modeling approaches that a document has rich information is not the case for microblogs (Zarrinkalam, 2015; Piao, 2016b).

Concept profiles. Concept-based user profiles are represented as conceptual nodes (concepts) and their relationships, and the concepts usually come from a pre-existing knowledge base (Gauch et al., 2007b). They can be useful for dealing with the problems that keyword profiles have. For example, WordNet (Miller, 1995) groups related words together in concepts called *synsets*, which has been proved useful for dealing with *polysemy* in other domains. For example, Stefani, 1998 used WordNet synsets for representing user interests in order to provide personalized website access instead of using keywords as they are often not enough for describing someone's interests. Another type of concept is *entities with URIs*. For instance, this involves using `wiki:Apple_Inc.` to denote the company `Apple`, which is disambiguated based on the context of the word *apple* in a text such as a

[23] http://www.opencalais.com/

tweet and linked to the corresponding entity in knowledge bases such as Wikipedia or DBpedia.

One of the advantages of leveraging Wikipedia/DBpedia entities is that we can exploit the background knowledge of these entities to infer user interests which might not be captured when using keyword-based approaches. For instance, a big fan of the `Apple` company would be interested in any brand-new products from `Apple` even if the names of these products have never been mentioned in the user's profiles (Lu et al., 2012). With the potential of inferring user interests using KBs, a large number of previous studies have used Wikipedia/DBpedia entities for building user interest profiles (Lu et al., 2012; Faralli et al., 2015b; Abel et al., 2011b; Abel et al., 2011a; Abel et al., 2011c). Similar to using Wikipedia/DBpedia entities, Ahn et al., 2012 leveraged Facebook entities (pages) for representing user interests.

Instead of using specific entities as mentioned above, *category-based* representation of user interests aims at using categories covering these entities for representing user interests. Take the following real-world tweet as an example (Michelson and Macskassy, 2010):

"#Arsenal winger Walcott: Becks is my England inspiration: http://tinyurl.com/37zyjsc",

there are four entities such as `wiki:Arsenal_F.C.`, and `wiki:Theo_Walcott` within the tweet, and the intersection categories of these entities such as `wiki:Category:English_Football_League` (see Figure 2.3) can be used for representing the topic of interests instead of the four entities.

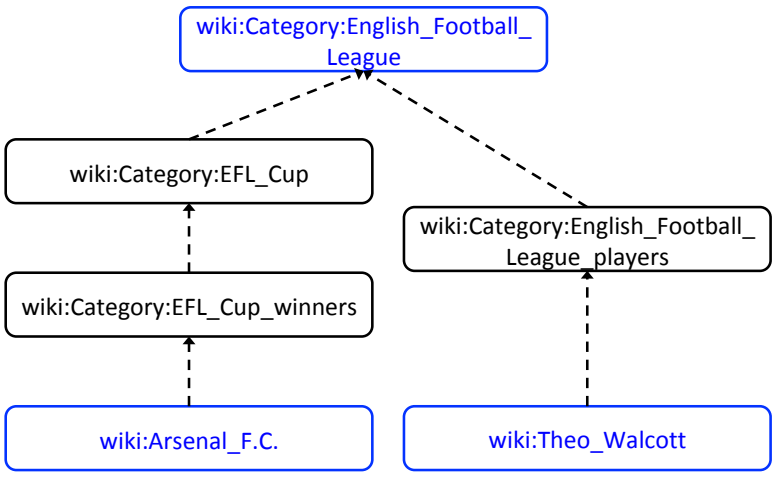


FIGURE 2.3: Example of an intersection category of two entities in Wikipedia.

Michelson and Macskassy, 2010 was one of the first studies using 23 top-level Wikipedia categories to represent user interest profiles. To build category-based user interest profiles, the first step is extracting entities from an information source (e.g., posts of users), which is similar to building *entity-based* user interest profiles. Subsequently, these entities have been used for obtaining their categories based on different proposed approaches. Kapanipathi et al., 2014 proposed representing user interest profiles as Wikipedia categories based on a hierarchical knowledge base. The knowledge base is a refined Wikipedia category system which is obtained by their proposed approach. Similar representations of user interests were proposed in Faralli et al., 2015b using WiBi (Wikipedia Bitaxonomy) (Flati et al., 2014) as the hierarchical knowledge base, and Faralli et al., 2017 with pruning approaches on Wikipedia categories. However, Budak et al., 2014 argued that Wikipedia categories tend to get out-of-date, and cannot keep up with the evolving topics on Twitter. Instead, the authors in Budak et al., 2014 leveraged categories in a taxonomy derived from the Open Directory Project (ODP[24]) to construct user interest profiles. For the comparison of *entity-* and *category-based* representations of user interests, Orlandi et al., 2012 investigated *entity-* and *category-based* user interest profiles aggregated from Twitter and Facebook, and evaluated those profiles based on a user study. The results suggested that profiles using DBpedia entities for representing user interests are more accurate than the profiles represented by DBpedia categories.

Entities and categories of Wikipedia/DBpedia can be seen as low and high level concepts. Some KBs in the form of a concept taxonomy, e.g., ACM Computer Classification System (CCS), may not distinguish entities and categories. The user model using concepts of KBs for representing user interests can be seen as an adapted *overlay model* in Intelligent Tutoring Systems, which aims to represent an individual user's knowledge as a subset of a *domain model* and reflects the expert-level knowledge of the subject (Brusilovsky and Millán, 2007). Different from the traditional overlay model, the adapted overlay model using concepts in KBs aims to represent an individual user's interests as a subset of the *cross-domain* background knowledge from a knowledge base.

Semantic network profiles. Semantic network-based profiles aim to address the polysemy problem of keyword-based profiles by using a weighted semantic network in which each node represents a specific word or a set of related words. This type of profile is similar to concept profiles in the sense of the representation of conceptual nodes and the relationships between them, despite the fact that the concepts in semantic network profiles are learned (modeled) as part of user profiles by collecting positive/negative feedback

[24] http://www.dmoz.org/, closed as of March 17, 2017

from users (Gauch et al., 2007b). As most previous works have focused on implicitly constructing user interest profiles in microblogging services, this type of profile has not been used in the domain of user modeling in microblogging services.

Although leveraging KG concepts for representing user interest profiles has the advantage of enhancing those profiles by using the background knowledge in a KG, it has some limitations as well. For example, KGs such as DBpedia do not cover all existing and emerging concepts in OSNs. Furthermore, most KGs lack full coverage for the lexicographic senses of lemmas, which can be provided by WordNet instead. In this regard, we propose a rich representation of user interest profiles which leverages both DBpedia concepts and WordNet synsets in Section 3.7.

Polyrepresentation of User Interest Profiles

Although it is common to use a single representation with respect to a user interest profile, the *polyrepresentation theory* (Ingwersen, 1994) based on a cognitive approach indicates that the overlaps between a variety of aspects or contexts with respect to a user within the information retrieval process can decrease the uncertainty and improve the performance of information retrieval. Based on this theory, White et al., 2009 studied polyrepresentation of user interests in the context of a search engine. The authors combined five different views/contexts of a user for inferring user interests, and showed that polyrepresentation is viable for user interest modeling.

Several studies have proposed constructing multiple user interest profiles in OSNs as well. For instance, the authors in Lu et al., 2012 and Chen et al., 2010 both constructed two user interest profiles for each user. In Chen et al., 2010, two keyword-based user interest profiles were built based on the tweets of a user and the tweets of the user's followees for recommending URLs on Twitter. Lu et al., 2012 proposed using Wikipedia entities and the affinity of other users to construct two user interest profiles. For a given user, the first user profile was represented as a vector of Wikipedia entities (articles), which were extracted from the user's tweets. Similar to Chen et al., 2010, the authors in Lu et al., 2012 also exploited the followees of target users to construct the second user profile. However, differing from Chen et al., 2010, the second interest profile is a vector with weights depending on the type of interaction between the user and their followees (retweet, reply or mention) with respect to the user.

User interest profiles can also include multiple views/aspects of a user. For example, Hannon et al., 2012 proposed a multi-faceted user profile which

includes user interests from target users, their followees, and followers. Figure 2.4 shows an example from Hannon et al., 2012 for representing user interests, where user interests are represented based on the tags associated with the list memberships of users, followees, or followers provided by a third-party service.

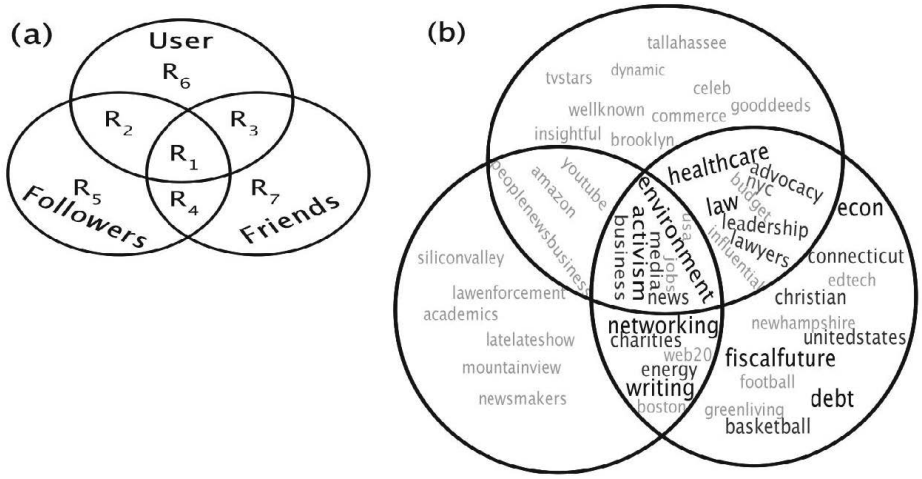


FIGURE 2.4: (a) Intensional and extensional profile regions. (b) Barack Obama's profile showing the tags associated with Obama and his followees (friends in the figure) and followers (Hannon et al., 2012).

In this thesis (Section 4.4), we also investigate the polyrepresentation of user interest profiles when the target users are *passive* users who do not generate content on OSNs but who keep following other people for consuming information. To infer user interests for passive users, we exploit the *biographies* and the *list memberships* of followees to construct two user interest profiles for them. The biographies of followees provide *self-descriptions* of themselves while the *list memberships* provide *others-descriptions* about them. We aim to investigate whether the polyrepresentation of user interest profiles based on those two different views with respect to users' followees improves user modeling performance or not.

2.3.3 Profile Construction and Enhancement

So far we have focused our discussion on collecting data from various sources for inferring user interests, and different representations for interest profiles. In the following, we provide details on how user interest profiles for a certain representation have been constructed based on the collected data. The overview of the construction and enhancement of user interest profiles is carried out based on three criteria: (1) *profile construction*, (2) *profile enhancement*, and (3) *temporal dynamics of user interests*.

Profile construction

Based on a defined representation of user interest profiles, a profile constructor aims to determine the weights of user interest units such as words or concepts in user profiles using a certain *weighting scheme*. A *weighting scheme* is a function or process to determine the weights of interest units where the weights denote the importance of these interests with respect to a user. For example, a common weighting scheme is using the frequency of an interest unit w_i (e.g., keyword) to denote the importance of w_i with respect to a user u, which can be formulated as below, when the data source is u's posts:

$$TF_u(w_i) = frequency\ of\ w_i\ in\ u's\ posts. \tag{2.1}$$

Despite its simplicity, this approach has been widely used in the literature, particularly in entity-based user interest representations (Kapanipathi et al., 2014; Abel et al., 2011c; Tao et al., 2012).

Information units such as entities extracted from tweets might come with their confidence scores (TC), which can be incorporated into a weighting scheme. In Jiang and Sha, 2015, the authors used TF with the confidence scores of information units extracted from tweets as their weighting scheme. Garcia Esparza et al., 2013 leveraged a classifier to obtain a set of ranked categories $C = \{c_1, \ldots, c_m\}$ with respect to each tweet including a URL by analyzing the content of the URL, and used a positional weighting scheme to measure the weight of each category as a TC with respect to a tweet. Peñas et al., 2013 also exploited the URLs mentioned in tweets to infer user interests. However, they leveraged the categories of URLs from OpenDNS[25] and DBpedia instead of exploring the content of these URLs, and the weights of categories were simply represented as 1 or 0 (interested or not).

TF·IDF is another common weighting scheme for weighting an interest unit from a user's posts. The IDF score of w_i with respect to a user u based on u's tweets can be measured as below (Chen et al., 2010):

$$IDF_u(w_i) = log\left[\frac{\#\ all\ users}{\#\ users\ using\ w_i\ at\ least\ once}\right]. \tag{2.2}$$

It is worth noting that IDF can also be applied after the *profile enhancement* process (e.g., Nishioka and Scherp, 2016). We use TF·IDF as the default

[25] OpenDNS cloud websites tagging, http://community.opendns.com/domaintagging/

weighting scheme for our concept-based user interest profiles after comparing it against with concept frequency (Section 3.4).

In Vu and Perez, 2013, the authors compared different weighting schemes such as TF·IDF, TextRank (Mihalcea and Tarau, 2004), and TI-TextRank which was proposed by the authors by combining TF·IDF and TextRank. The evaluation based on a user study showed that TI-TextRank performs best for ranking keywords from the tweets of users.

Instead of weighting interest units appearing in users' posts, some approaches extracted interest units such as entities by measuring the similarity between a post and an interest unit. For instance, Lu et al., 2012 and Narducci et al., 2013 used the Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch, 2007) algorithm, which is designed to compute the similarity between texts, for obtaining the weights of entities for each tweet of a user. Those weights of entities were then aggregated for constructing entity-based primitive interests of users. Ahn et al., 2012 quantified the degree of an interest unit, i.e., a Facebook entity, based on two factors: (1) the familiarity with each social neighbor, and (2) the similarity between the topic distributions of a piece of social content and an interest unit. *Social content* is the combined text of a post and the user comments associated with it.

The weights of user interests have also been learned in unsupervised ways in the literature. For instance, Weng et al., 2010 treated tweet histories of each user as a big document, and used LDA to learn topic distributions for each user. In Xu et al., 2011, the authors proposed a modified author-topic model (Rosen-Zvi et al., 2004) for distinguishing interest-related and unrelated tweets for learning topic distributions of users. Budak et al., 2014 proposed a probabilistic generative model to infer user interest profiles which are represented as an interest probability distribution over ODP categories. In their proposed approach, the authors considered three aspects such as (1) the posts of a target user, (2) the activeness of the user, and (3) the influence of friends. They assumed that time is divided into fixed time steps, and transformed the problem into inferring the probability of a user being interested in each of the interests, given a social network that evolves over time, including posts and social network information. Sang et al., 2015 also proposed a probabilistic framework for inferring user interest profiles. Different from Budak et al., 2014, Sang et al., 2015 assumed users have long- and short-term interest (topic) distributions. Long-term interests denote the stable preferences of users while short-term interests denote user preferences over short-term topics during some events in OSNs. However, they did not consider the social networks of users.

In Zarrinkalam and Kahani, 2015, user interest profiles were represented as

topic vectors where each topic is a set of temporally correlated entities on Twitter. To this end, an entity graph based on their temporal correlation as defined by the authors was constructed, and the topics in a time interval were extracted using some existing community detection algorithms such as the *Louvain* method (Rotta and Noack, 2011). Subsequently, each topic z was transformed into a set of weighted entities using the *degree centrality* of an entity in the topic (community).

For user interest profiles exploiting social networks such as followees, there have been various methods proposed in previous studies (Chen et al., 2010; Lu et al., 2012). For example, Lu et al., 2012 constructed a user interest profile based on a user affinity vector consisting of weights with respect to other accounts the user has interacted with. The weights were calculated by considering explicit interactions between users based on follower/followee information, and explicit interactions such as the number of tweets that are a reply, retweet, or mention between two users.

Chen et al., 2010 built two keyword-based user profiles for a user u; one is based on u's tweets, and the other is based on the tweets of u's followees. The TF·IDF weighting scheme was used for constructing keyword-based user profiles based on u's tweets, which are called *self-profiles*. To build a user profile based on followees' tweets, the authors first retrieved a set of *high-interest words* for followees as follows: For each *self-profile* for followees of u, they picked all words that have been mentioned at least once, and selected the top 20% of words based on their occurrences. In addition, the words that are not in other followees' profiles were removed. Subsequently, the weight of each word in the set of *high-interest words* was measured as below:

$$\begin{aligned} FTF_u(w_i) = & \#\ u's\ followees\ who\ have\ w_i \\ & as\ one\ of\ their\ high-interest\ words. \end{aligned} \tag{2.3}$$

A similar approach to $FTF_u(w_i)$ was adopted in Bhattacharya et al., 2014. The authors in Bhattacharya et al., 2014 explored the list memberships of followees to extract topics of interests for a target user, where the weight of a keyword is measured by the number of followees who have that keyword in their list memberships. Motivated by $FTF_u(w_i)$, we propose and evaluate a similar weighting scheme for weighting the entities extracted from the list memberships of users' followees (Section 4.3.1).

Profile enhancement

The constructed primitive user interest profiles such as the ones represented by entities, can be further enhanced using external knowledge to deliver the final interest profiles. The approaches used in the literature for enhancing primitive user interests have mainly leveraged *hierarchical* or *graph-based* knowledge.

Leveraging hierarchical knowledge. Knowledge bases such as Wikipedia have been widely used for enhancing user interest profiles. For example, Kapanipathi et al., 2014 proposed representing user interest profiles as Wikipedia categories based on a hierarchical knowledge base, which is a refined Wikipedia category system built by the authors. The user interest profiles were constructed using the hierarchical knowledge base with the following two steps. First, Wikipedia entities in users' tweets were extracted as their primitive interests. Second, these entities were used as activated nodes to apply an adapted *spreading activation* (Collins and Loftus, 1975) function on the hierarchical knowledge base.

The spreading activation function proposed by Kapanipathi et al., 2014 can be applied to any case where a set of entities and a hierarchical knowledge base are available. Therefore, many studies that followed have adopted this spreading activation function but with different approaches for extracting entities or with different hierarchical knowledge bases (Besel et al., 2016b; Besel et al., 2016a; Piao and Breslin, 2017b; Nishioka and Scherp, 2016; Große-Bölting et al., 2015). For instance, Nishioka and Scherp, 2016 extracted entities and applied the spreading activation function on STW, which is a hierarchical knowledge base from the economics domain. Große-Bölting et al., 2015 investigated several spreading activation functions ranging from a basic one (see Equation 2.4 where D denotes the decay factor) to the one proposed in Kapanipathi et al., 2014, applied on the ACM CCS concept taxonomy in the computer science domain.

$$a_t(j) \leftarrow a_{t-1}(j) + D \times a_{t-1}(i) \tag{2.4}$$

The results showed that using the basic spreading activation function provided the best user interest profiles compared to using other ones in the context of research article recommendations.

In Besel et al., 2016b and Besel et al., 2016a, the authors extracted entities by mapping followees' Twitter accounts to Wikipedia entities, and used WiBi (Flati et al., 2014) as their hierarchical knowledge base for applying the spreading activation function proposed in Kapanipathi et al., 2014. Similarly, Faralli et al., 2015b also mapped followees' Twitter accounts to Wikipedia entities, and used them as users' primitive interests for propagation with

WiBi. However, a simpler propagation strategy was adopted in Faralli et al., 2015b. In Faralli et al., 2017, the authors extended their previous work (Faralli et al., 2015a) and proposed a methodology to build *Twixonomy*, which is a Wikipedia category taxonomy. *Twixonomy* is built by using a graph pruning approach based on a variant of Edmonds optimal branching (Edmonds, 1968). The authors showed that the proposed approach can generate a more accurate taxonomy compared to the approach proposed in Kapanipathi et al., 2014. One issue with these approaches mapping followees' accounts to Wikipedia entities is that only a limited percentage of followees' accounts can be mapped to corresponding entities. For example, only 12.7% of followees' accounts can be mapped to Wikipedia entities as described in Faralli et al., 2015b.

Instead of using refined hierarchical knowledge from Wikipedia/DBpedia, some studies have explored other types of hierarchical knowledge bases as well. Kang and Lee, 2016 proposed mapping news categories to tweets for constructing user interest profiles. The authors leveraged news categories from two popular news portals in South Korea (Naver News[26] and Nate News[27]) to build their category taxonomy. This taxonomy consists of 8 main categories and 58 sub-categories, and each category consists of all news articles in the two news corpuses. To assign categories to a tweet, each tweet and news category are represented as a term vector where the weights of terms are calculated using TF·IDF first. As there might be a semantic gap between terms in social media and news portals, the authors leveraged Wikipedia to resolve the problem. Three different approaches such as *entity-*, *category-*, and *category cluster*-based methods were proposed in order to transform the term vectors of tweets and news categories into the same vector space. The results showed that the category-based approach for assigning news categories to a given tweet provides the best performance compared to *entity-*, and *category cluster-based* methods. The category-based approach transforms the term vectors of tweets and news categories into Wikipedia category vectors, and assigns the top two news categories to each tweet based on the cosine similarity between their two Wikipedia category vectors. These news categories of a user's tweets are then aggregated to construct the final user interest profiles. In addition, the author also showed that combining these three approaches to measure the similarity between a news category and a tweet can further improve the accuracy.

Jiang and Sha, 2015 leveraged external knowledge sources such as DBpedia, Freebase (Bollacker et al., 2008), and Yago (Suchanek et al., 2007) for constructing a Topic Hierarchy Tree (THT), which is a hierarchical knowledge

[26] http://news.naver.com/
[27] http://news.nate.com//

base that consists of over 1,000 topics distributed in 5 levels. However, the details for obtaining the THT were not discussed in their study. In Bhargava et al., 2015, the authors manually built a category taxonomy based on Facebook Page categories and the Yelp[28] category list. The category taxonomy in Bhargava et al., 2015 consists of three levels with 8, 58, and 137 categories in each level, respectively. The authors used features such as entities, hashtags, and document categories which can be extracted from Facebook *likes* and UGC as users' primitive interests, and then measured the confidence of each concept in the category taxonomy based on these features using a Semantic Textual Similarity system (Han et al., 2013).

Leveraging graph-based knowledge. Instead of leveraging hierarchical knowledge, many studies have leveraged graph-based knowledge for enhancing user profiles. For example, Michelson and Macskassy, 2010 exploited Wikipedia categories directly for propagating a user's primitive interests. The authors summed the scores of a category which appeared at multiple depths in the category graph. Differing from exploring the categories of a specified depth (Michelson and Macskassy, 2010), Siehndel and Kawase, 2012 represented user interest profiles using 23 top-level categories of the root node `Category:Main_Topic_Classifications` in Wikipedia. The Wikipedia entities in users' tweets were extracted as their *primitive interests*, and these entities were then propagated up to the 23 top-level categories with a discounting strategy for the propagation.

With the advent of large, cross-domain KGs such as DBpedia, different approaches leveraging background knowledge from KGs have been investigated. A knowledge graph is a knowledge base which consists of an ontology and instances of the classes in the ontology (Färber et al., 2015). The difference between a hierarchical category taxonomy such as WiBi and a knowledge graph such as DBpedia is that DBpedia goes beyond just categories to related entities via the entity's predicates. Depending on the propagation strategies for those entities in a user's primitive interests, different aspects, e.g., *related entities*, *categories* or *classes* of the entities can be leveraged for the propagation. For example, Peñas et al., 2013 enriched categories in users' primitive interests using similar categories defined by the `categorySameAs` relationship in DBpedia. Abel et al., 2012 proposed using background knowledge from DBpedia for propagating user interest profiles with respect to Points Of Interest (POI). The authors considered entities that were two hops away from a user's primitive interests and that were related to places. However, this approach did not consider any discounting strategy for the weights of propagated user interests. In Orlandi et al., 2012, the authors leveraged DBpedia categories one hop away from of the entities in a

[28] https://www.yelp.com/

user's primitive interests using a discounting strategy for propagating user interests.

Lu et al., 2012 exploited a Wikipedia entity graph to enhance the entity-based primitive interests. Compared to the DBpedia graph, where the edges between two entities are predefined predicates in an ontology, the edges in the Wikipedia entity graph denote the mentions of other entities in a Wikipedia entity/article. In contrast to exploiting Wikipedia categories, the intuition behind this approach is that if a user is interested in `IPhone`, the user might be interested in other products of `Apple`, instead of being interested in other mobile phones in the same category such as `Smartphones`. To this end, the authors used the ESA algorithm to extract entities from the tweets of users as their primitive interests, and then expanded these entities using a random walk on the Wikipedia entity graph.

Although some of the previous studies in the literature have explored DBpedia for enhancing the primitive interests of users, they have focused mainly on leveraging the *categories* of entities. In contrast, we investigate user modeling strategies leveraging several aspects of DBpedia, e.g., *related entities*, *categories* or *classes* of the entities for enhancing user interest profiles in Section 3.5.1.

Temporal dynamics of user interests

User interests in OSNs can change over time, and many studies have been conducted in order to investigate the temporal dynamics of user interests in OSNs. For example, Jiang and Sha, 2015 showed that, the similarity of current user interest profiles with the profiles at the beginning of the observation period of their dataset is the lowest while the similarity of current profiles with the ones built last month is the highest. Similarly, Abel et al., 2011b showed that a user interest profile built in an earlier week differs more from the current profile compared to the one built recently. The authors also showed that the *weekday* and *weekend* profiles have bigger differences compared to *day* and *night* profiles for each user based on their historical tweets.

In order to incorporate temporal dynamics of user interests into user modeling strategies, there are mainly two types of approaches: (1) *constraint-based* approaches, and (2) *interest decay functions*. The former one extracts user interest profiles based on specified constraints, e.g., using a *temporal constraint* to build user interest profiles based on their tweets posted in the last two weeks or using an *item constraint* to construct user profiles based on the last 100 tweets of the users. Compared to constraint-based approaches, interest

decay functions build user profiles with lower weights for older interests and higher weights for recent ones.

Constraint-based approaches. Abel et al., 2011b investigated several temporal constraints in their user modeling strategies on Twitter for a news recommender system. For instance, they considered *long-* and *short-term* profiles, and *weekend* profiles in the design space of user modeling. However, based on different user profile representations, e.g., using *topic-* or *entity-based* profiles, different results were observed. For example, long-term profiles outperform short-term profiles in terms of entity-based profiles while short-term profiles outperform long-term profiles in terms of topic-based user profiles. Similarly, although weekend profiles constructed based on the tweets posted on weekends only in a user's history outperform long-term profiles in the case of entity-based user profiles, the opposite result was observed for topic-based user profiles. Overall, entity-based user profiles considering the temporal dynamics perform best compared to topic-based user profiles. Nishioka and Scherp, 2016 compared both constraint-based approaches and interest decay functions for constructing user interest profiles on Twitter in the context of publication recommendations. Differing from the results in the domain of news, the results from Nishioka and Scherp, 2016 showed that a constraint-based approach constructing user interest profiles within a certain period performs better than using an interest decay function in the context of publication recommendations.

Interest decay functions. Instead of constructing user interest profiles in a certain period (e.g., short-term), or based on temporal patterns (e.g., weekends), many studies applied interest decay functions to long-term profiles. The intuition behind those interest decay functions is that a higher weight should be given to recent interests than old ones. Different types of interest decay functions have been proposed in previous studies such as a time-sensitive interest decay function proposed by Abel et al., 2011a or an exponential decay function proposed by Orlandi et al., 2012. For example, a popular interest decay function from Orlandi et al., 2012 is defined as follows:

$$(x(t)) = x_0 e^{-t/\beta}. \tag{2.5}$$

Here, $x(t)$ is the decayed weight at time t, and x_0 denotes the initial weight (at time $t = 0$). β is a parameter which controls the speed of exponential decay. This interest decay function also has an initial time window (7 days), and the interests in the time window are not discounted. The authors in Orlandi et al., 2012 set $\beta = 360 days$ and $\beta = 120 days$ for their experiment, and showed that using $\beta = 360 days$ performs better than using $\beta = 120 days$

in terms of an evaluation based on a user study. A similar decay function was used in Bhargava et al., 2015 and Nishioka and Scherp, 2016, where the weight of the last update was used instead of initial weight (Bhargava et al., 2015). In O'Banion et al., 2012, the authors also used an exponential decay function: $x(t) = x_0 0.9^d$ where d is the difference in days between the current date and the date that a concept was mentioned.

Despite those approaches that have been proposed for incorporating the temporal dynamics of user interests, their comparative performance in user modeling is not investigated. Hence, we provide a comparative study on the performance of different approaches for incorporating the temporal dynamics of user interests in Section 3.6.

2.3.4 Evaluation of the Constructed User Profiles

The final step for user modeling in OSNs is how to evaluate the constructed user interest profiles. Overall, there are three different approaches for evaluating the constructed user interest profiles: (1) evaluation based on a user study, (2) evaluation in terms of application performance, and (3) manual analysis of the constructed profiles.

Evaluation based on a user study

The first evaluation approach is based on a user study. This approach requires recruiting users for the experiment of building user interest profiles with their OSN accounts. Finally, these users provide feedback on the user interest profiles constructed by different user modeling strategies. For example, Narducci et al., 2013 evaluated user interest profiles built for 51 users from Facebook and Twitter based on their feedback on a 6-point discrete rating scale. Kapanipathi et al., 2014 recruited 37 users and built category-based user interest profiles based on their tweets on Twitter. Afterwards, the 37 users provided explicit feedback, e.g., Yes/Maybe/No with respect to the categories in those profiles. Similar approaches have been used in Bhattacharya et al., 2014, Besel et al., 2016a; Besel et al., 2016b, Budak et al., 2014, and Orlandi et al., 2012. However, instead of recruiting volunteers for an experiment, the authors in Budak et al., 2014 first inferred user interest profiles for 500 randomly chosen users with email addresses on Twitter, and emailed them using the email addresses in their profiles to get feedback about their inferred interests.

In Chen et al., 2010 and Nishioka and Scherp, 2016, the authors also conducted a user study but with respect to a specific application, i.e., a URL

recommender system on Twitter. Therefore, instead of directly giving feedback on the constructed user interest profiles, the users that participated in the study were given URL recommendations, and marked each URL as one of their interests or not. Instead of using the feedback of target users for the evaluation of inferred user interest profiles, Kang and Lee, 2016 and Michelson and Macskassy, 2010 labeled user interests by themselves or used other recruited annotators. Garcia Esparza et al., 2013 implemented a stream filtering system where users are represented based on 18 defined categories such as `Music` and `Sports`. For evaluation, the authors asked each participant to give explicit feedback on their profiles by deleting or adding categories that they felt were incorrect or missing.

Evaluation approaches based on the explicit feedback of profiled users would arguably be the most direct and accurate way for evaluating the inferred user interests of these users. However, they also require the recruitment of volunteers and impose an extra burden for users, and therefore limits the number of participants for evaluation (e.g., 37 users for evaluation in Kapanipathi et al., 2014).

Evaluation in terms of application performance

To evaluate the quality of inferred user interest profiles without imposing an extra burden to users, offline evaluation in terms of the performance of specific applications has been used. In this case, user interest profiles are used as an input to an application, such as a news recommender system where these profiles play an important role. Afterwards, different profiles created by different user modeling strategies are compared in terms of the application performance using each profile. For instance, when we evaluate different user modeling strategies in terms of a recommender system, we can adopt well-established evaluation metrics for RSs such as Normalized Discounted Cumulative Gain (nDCG) and Mean Reciprocal Rank (MRR).

Abel et al., 2011b evaluated three different user modeling strategies in the context of news recommendations. Sang et al., 2015 also evaluated user interest profiles in terms of news recommendations in addition to tweet recommendations. In Faralli et al., 2015b, the authors evaluated user interest profiles in terms of user classifications and recommendations. For the classification task, the user interest profiles were used for classifying each user to the appropriate label, e.g., Starbucks fan. For the recommendation task, the authors evaluated the performance of leveraging different hierarchical levels of interests with respect to interest recommendations using itemset mining.

Manual analysis of constructed profiles

There are other evaluation approaches used in some studies besides the aforementioned two methods. For example, Abel et al., 2011c compared the number of distinct entities and topics in user interest profiles in order to evaluate news-based enrichment of their tweets. In Faralli et al., 2017, the authors ran two experiments to evaluate their approach of building interest taxonomies. First, they compared their approach against other approaches proposed for constructing user interest taxonomies using other gold standard taxonomies. Second, they provided some samples of generated user interest profiles, and analyzed the final interests with respect to those users. Similarly, Xu et al., 2011 evaluated their topic modeling approach by comparing against other topic modeling methods in terms of *perplexity*, and then discussed some user interest profiles produced by different approaches.

In this thesis, we adopt the offline evaluation strategy which evaluates different user modeling strategies in terms of the performance of specific applications.

2.4 LOD-enabled Recommender Systems

In this section, we discuss LOD-enabled recommender systems which are useful for addressing the second and third research challenges identified in Section 1.1.

The term Web of Data, often referred to as the Semantic Web, Web 3.0 or Linked Data, indicates a new generation of technologies responsible for the evolution of the current Web from a Web of interlinked documents to a Web of interlinked data (Heath and Bizer, 2011). The goal is to discover new knowledge and value from data, by publishing them using Web standards (primarily RDF[29]) and by enabling connections between heterogeneous datasets. In particular, the term Linked Open Data denotes a set of best practices for publishing and linking structured data on the Web. The project includes dozens of RDF datasets interlinked with each other to form a giant global graph, the so called Linked Open Data cloud[30] (see Figure 2.5).

DBpedia is a first citizen in this cloud since it represents the nucleus of the entire LOD initiative (Auer et al., 2007). It is the semantic representation of Wikipedia and it has become one of the most important and interlinked

[29] Resource Description Framework (RDF), https://www.w3.org/RDF/
[30] http://lod-cloud.net/

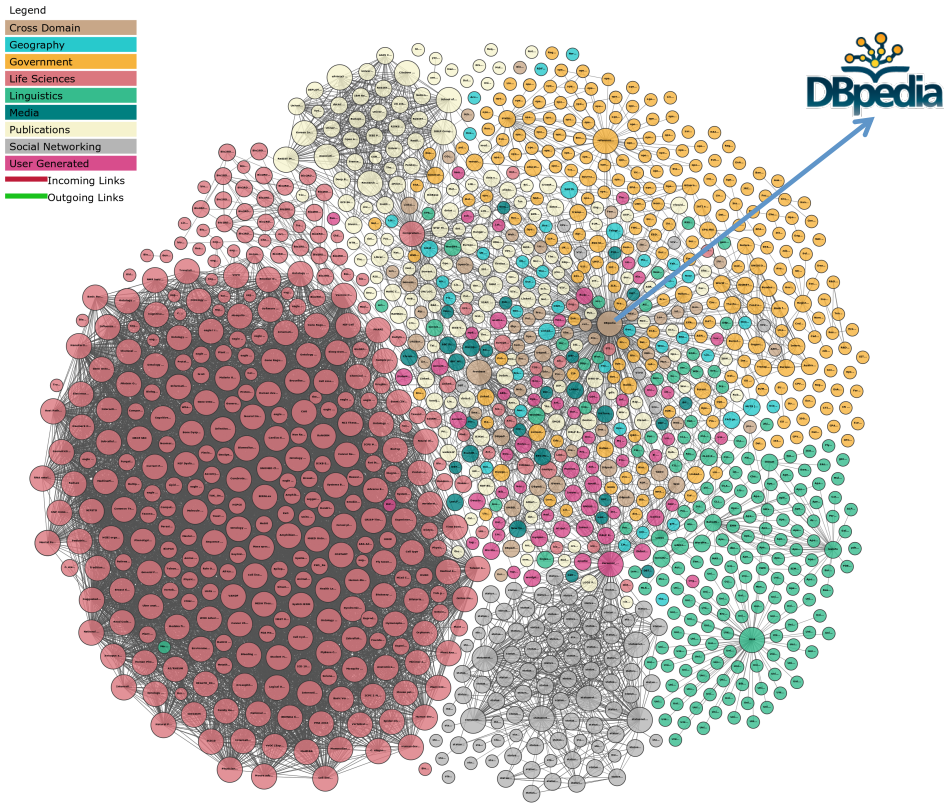


FIGURE 2.5: Linked Open Data cloud diagram (Abele et al., 2017).

datasets on the Web of Data. Compared to traditional taxonomies or lexical databases (e.g. WordNet) it provides a larger and "fresher" set of terms, continuously updated by the Wikipedia community and integrated into the Web of Data. The latest version of DBpedia describes 4.58 million things, including 1,445,000 persons, 735,000 places, 411,000 creative works such as music albums, films and video games, 241,000 organizations, 251,000 species and 6,000 diseases[31].

This cross-domain background knowledge about entities is freely accessible via its SPARQL endpoint[32]. For example, Figure 2.6 shows pieces of background knowledge about the movie `dbr:The_Godfather` in RDF triples, which can be obtained from DBpedia. A RDF triple consists of a subject, a predicate and an object. As we can see from the figure, there can be incoming knowledge, e.g., `dbr:Carlo_Savina` → `dbo:knownFor` → `dbr:The_Godfather` where `dbr:The_Godfather` is used as an object, as well as outgoing knowledge such as `dbr:The_Godfather` → `dbo:director` → `dbr:Francis_Ford_Coppola` where `dbr:The_Godfather` is a subject.

[31] http://wiki.dbpedia.org/about
[32] http://dbpedia.org/sparql

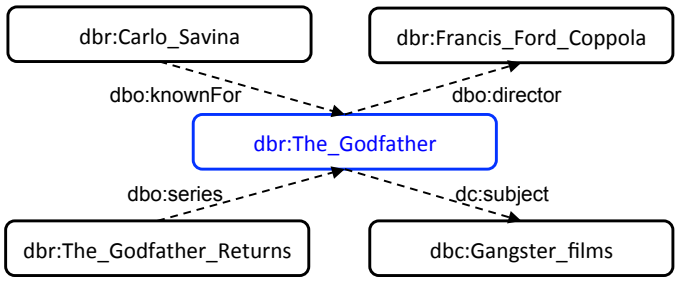


FIGURE 2.6: An example of background knowledge about the movie "*The Godfather*" from DBpedia.

In the context of the great amount of freely accessible information, many studies have been conducted in order to consume the knowledge provided by LOD for adaptive systems such as recommender systems (Di Noia et al., 2014; Gemmis et al., 2015a). In the following, we review related work on LOD-enabled recommender systems.

2.4.1 Semantic Similarity/Distance Measures

Maedche and Zacharias, 2002 defined a set of similarity measures for comparing ontology-based metadata by considering different aspects of an ontology separately. They propose differentiating across three dimensions for comparing two resources: taxonomic, relational and attribute similarities. However, the similarity measures depend on some strong assumptions about the model such as "Ontologies are strictly hierarchical such that each concept is subsumed by only one concept", which is not the case in terms of many KGs such as DBpedia.

The first attempts to leverage LOD for recommender systems were by Heitmann and Hayes, 2010 and Passant, 2010b. Heitmann and Hayes, 2010 proposed a framework using LOD for open collaborative recommender systems. Passant, 2010b proposed a measure named *LDSD* (Linked Data Semantic Distance) to calculate semantic distance on Linked Data. The distance measure considers direct links from entity A to entity B and vice versa. In addition, it also considers the same incoming and outgoing nodes via the same predicates of entities A and B in a graph. The distance measure has a scale from 0 to 1, where a larger value denotes less similarity between two entities. In later work, the author used the *LDSD* similarity measure in a recommender system based on DBpedia entities which recommends similar music artists based on the artists in a user's preference profile (Passant, 2010a).

Leal et al., 2012 proposed a similarity measure for computing the semantic relatedness of entities in DBpedia. The proposed similarity measure is based on a notion of *proximity*, which measures how connected two entities are, rather than how distant they are. This means that the similarity measure considers both distance and the number of paths between two nodes. This similarity measure extends each step to find longer paths between two entities, and penalizes proximity by steps, i.e., a longer path contributes less to the proximity. The extension is terminated by a defined value of maximum steps (max step). The similarity measure is implemented in a tool named "Shakti", which extracts an ontology for a given domain from DBpedia and uses it to compute the semantic relatedness of entities. However, the authors did not consider incoming nodes (entities) and the predicates of these entities as *LDSD* did. In addition, the weights assigned to predicates are defined manually and the authors pointed out the need for an automated approach as future work. We use *Shakti* to refer to this measure in the rest of the thesis. Based on the *Shakti* measure, Strobin and Niewiadomski, 2013 propose a method to find the weights automatically by using a genetic optimization algorithm based on a training dataset from Last.fm[33]. This method is quite efficient at learning the weights automatically. However, it needs a gold standard dataset (e.g., the Last.fm dataset for the music domain) to learn the weights of predicates which is not always available in other domains.

More recently, Alfarhood et al., 2017a proposed a semantic distance measure which considers the connected entities beyond the ones one or two hops away in *LDSD*. In a later work, Alfarhood et al., 2017b proposed another distance measure which applies link differentiation strategies for measuring the linked data semantic distance between two entities in a linked dataset such as DBpedia. In contrast to distance-based approaches, Meymandpour and Davis, 2016 proposed *PICSS* (Partitioned Information Content-based Semantic Similarity), which is an information content-based semantic similarity measure for measuring the similarity between two entities. *PICSS* is a feature-based similarity measure since it derives the feature vectors of entities, and then applies a weighted Jaccard similarity to the feature vectors for measuring the similarity between two entities.

For evaluation, every work proposed its own evaluation method for its measure, and none of these studies have compared their proposed similarity measures to others. For example, some have evaluated the similarity measures in terms of specific domains of recommender systems (Passant, 2010a; Passant, 2010b; Groues et al., 2012; Leal et al., 2012) while others have evaluated them in terms of clustering problems (Maedche and Zacharias, 2002). In Chapter 5, we propose a semantic similarity measure, and evaluate it by

[33] http://last.fm

comparing it against *LDSD* and *Shakti*. These semantic similarity/distance measures have been designed to work directly on LOD without considering the collaborative view of users, and therefore can be used for the initial state of a recommender system when there exists only a few liked items for each user.

2.4.2 Using Graph-based Algorithms

Based on the nature of the graph structure of DBpedia, *graph-based* approaches have been proposed for LOD-enabled recommender systems. For example, Ostuni et al., 2014 proposed a neighborhood-based graph kernel to measure the semantic similarity between two entities. The item/entity similarities were computed based on their neighbor entities in the local neighborhood graph of each entity. However, this approach considered the DBpedia graph as an undirected and homogeneous graph. In Nguyen et al., 2015, the authors investigated two existing similarity metrics, *SimRank* (Jeh and Widom, 2002) and a *personalized PageRank* algorithm (Haveliwala, 2003), in order to compute the similarity between entities in RDF graphs, and their usage to feed a content-based recommender system. Similar to Ostuni et al., 2014, both *SimRank* and *PageRank* were designed for measuring similarity in homogeneous graphs, which is not the case of DBpedia.

Musto et al., 2016b proposed combining the background knowledge about items and user-item interactions into a single graph, and then applying graph-based algorithms such as personalized PageRank. Figure 2.7 (Musto et al., 2016b) shows a portion of a combined graph, which consists of user-item interactions (i.e., item preferences of users) and the background knowledge with respect to items in DBpedia. As the computational complexity of the personalized PageRank algorithm grows by incorporating the background knowledge about items from DBpedia, the authors further investigated several LOD feature selection strategies. Their experimental results showed that most LOD-based predicates in the movie domain are relevant, i.e., most of the feature selection strategies provide their best performance with higher number of features. In contrast, the best-performing configurations in the book domain leverage 10 features, which shows that the knowledge of book entities from DBpedia is noisy. In a later work (Musto et al., 2017), the authors further showed that a proper tuning of personalized PageRank parameters with a better weighting distribution strategy for the enriched information of items can improve the recommendation performance. Similar to other graph-based approaches, they have to treat the combined graph as a homogeneous one in order to apply the personalized PageRank algorithm.

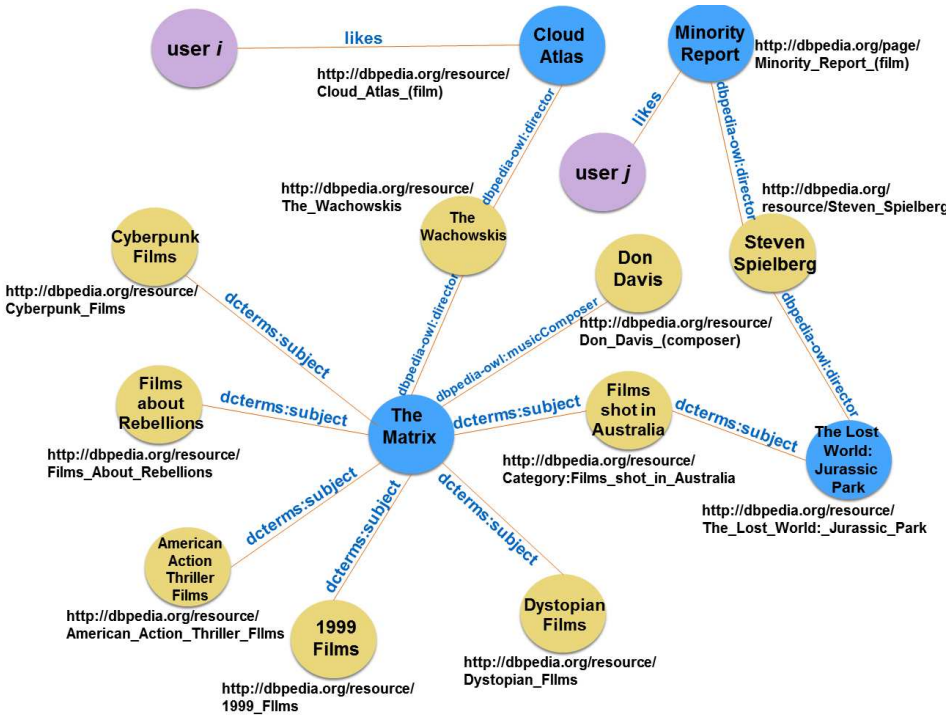


FIGURE 2.7: A portion of a combined graph which consists of user-item interactions and the background knowledge about items encoded in DBpedia (Musto et al., 2016b).

2.4.3 Machine Learning Approaches

Peska and Vojtas, 2013 used a content boosted matrix factorization approach (Forbes and Zhu, 2011) with background knowledge of books from DBpedia for providing book recommendations. The authors treated each item as a subject in DBpedia and extracted all predicate-object pairs with respect to items as boolean features. Di Noia et al., 2012a adapted the Vector Space Model (VSM) to a LOD-based setting, and represented the whole RDF graph as a matrix. On top of the VSM representation, they used the Support Vector Machine (SVM) as a classifier to predict if a user would like an item or not. Using the same representation, they also proposed assigning a weight to each predicate that represents its worth with respect to the user profile (Di Noia et al., 2012b). In this regard, they used a Genetic Algorithm (GA) to learn the weights of predicates that minimize the misclassification errors.

More recently, Noia et al., 2016; Ostuni et al., 2013 proposed *SPRank*, which is a semantic path-based approach using learning-to-rank algorithms. This approach first constructed a graph based on user-item interactions and the background knowledge of items from LOD. Afterwards, features, called *semantic paths*, were extracted based on the number of paths between a user and an item with min-max normalization. The extracted features were then

fed into existing learning-to-rank algorithms such as *LMART* (Wu et al., 2010) provided by RankLib[34]. This approach is also a graph-based approach, and the common requirement for *graph-based* approaches including *SPRank* is that a combined graph such as the one in Figure 2.7 has to be built based on user-item interactions and background knowledge from LOD.

There have also been some other interesting directions related to LOD-enabled recommender systems such as the practical LODRecSys (Oliveira et al., 2017), explaining recommendations using LOD (Musto et al., 2016a), rating predictions based on matrix factorization with semantic categories (Rowe, 2014), and cross-domain recommendations (Heitmann and Hayes, 2014; Heitmann, 2012). For example, Oliveira et al., 2017 presented a recommender system in the movie domain that consumes LOD (not restricted to DBpedia), which was evaluated by comparing it to seevl (ISWC challenge winner at 2011). Different types of evaluation metrics have been used such as accuracy, novelty etc. The authors from Musto et al., 2016a presented ExpLOD - a framework which can generate explanations in natural language based on the LOD cloud. Musto et al., 2016b investigated various feature (predicate) selection strategies and their influences on recommendation performance in terms of accuracy and diversity in the movie and book domains. Lalithsena et al., 2016 proposed a novel approach using *type-* and *path-based* methods to extract a subgraph for domain specific recommendation systems. They presented that their approach can decrease 80% of the graph size without losing accuracy in the context of recommendation systems in the movie and book domains. Figueroa et al., 2015, Figueroa et al., 2017 and Gemmis et al., 2015b also provide detailed reviews for LOD-enabled recommender systems.

Although various types of approaches have been explored for LOD-enabled recommender systems, factorization machines (Rendle, 2010), which is a state-of-the-art factorization model framework has not been investigated with LOD-enabled features. In Section 6.2, we propose *LODFM* which leverages different sets of lightweight LOD-enabled features for factorization machines, and evaluate it with a comparison of other baseline recommendation approaches including *SPRank*.

Despite the fact that KGs provide billions of machine-readable facts about entities, they are far from complete (Galárraga et al., 2017), and a dedicated line of research has focused on the task of KG completion (Franz et al., 2009; Drumond et al., 2012). Most previous studies as well as *LODFM* do not incorporate the incompleteness of KGs as they leverage the existing knowledge about items. In Section 6.3, we leverage a co-factorization model

[34] https://sourceforge.net/p/lemur/wiki/RankLib/

to investigate transfer learning (Pan and Yang, 2010) between these two tasks: (1) *item recommendations*, and (2) *KG completion* with respect to the domain of items. In contrast to *multitask learning*, which aims to optimize objective functions in several tasks, *transfer learning* aims to optimize the objective function of a "target" task by transferring knowledge from a "source" task.

2.5 Research Challenges Tackled in This Thesis

In the previous sections of this chapter, we discussed the background work related to user modeling and recommender systems in OSNs. The rest of the thesis will propose semantics-aware user modeling and recommendation approaches in the context of OSNs in order to tackle our research questions which can be summarized as follows.

Semantics-aware user modeling on microblogging social networks

Although there have been many inspiring works related to semantics-aware user modeling strategies (Orlandi et al., 2012; Abel et al., 2012; Kapanipathi et al., 2014; Besel et al., 2016a), different aspects of KGs and various types of user activities in OSNs were not fully explored for inferring user interest profiles in the literature. In addition, those dimensions of user modeling discussed in Section 2.3 have been studied separately, and there is a lack of research on the synergistic effect of those dimensions for user modeling.

- (How) can we leverage different aspects of knowledge graphs to infer and represent user interest profiles from different types of user activities on microblogging social networks?
- (How) can we incorporate different user modeling dimensions such as the temporal dynamics of user interests in order to construct better user interest profiles?

In **Chapters 3** and **4** we will address these research questions, and propose several user modeling strategies for inferring user interest profiles with respect to *active* and *passive* users. To this end, we first investigate each user modeling dimension separately (Section 3.3-3.6), and then provide a study on comprehensive user modeling strategies by combining various dimensions together (Section 3.8), which has not been studied in the literature. Furthermore, we propose user modeling strategies for passive users by investigating different types of user activities (beyond the creation of posts) to infer their interest profiles in **Chapter 4**.

Semantic similarity measures for recommending items in cold-start scenarios

There exists several semantic similarity/distance measures with the aim of providing item recommendations, such as "If you like X, you should like Y", based on direct and indirect links that exist between two items/entities in KGs (Passant, 2010b; Leal et al., 2012; Groues et al., 2012; Strobin and Niewiadomski, 2013). However, each study applied its own evaluation strategy, and lacks comparison with other similarity measures. Also, there exists little research on the effect of the knowledge sparsity with respect to items in KGs for recommendations based on those similarity/distance measures.

- How can we improve the performance of *LDSD* by resolving some limitations of it?
- Do different sparsities of background knowledge from KGs with respect to items affect the performance of recommendations based on semantic similarity/distance measures?

We will answer these research questions in **Chapter 5**, and propose a semantic distance measure which measures the similarity between two items/entities and resolves some limitations of *LDSD*. In addition, we study "linked data sparsity" and its effect on recommendations made by semantic distance measures.

Semantics-aware machine learning approaches for item recommendations

Most previous studies require increased effort to maintain an additional graph based on user-item interactions and background knowledge about items from LOD in their approaches (Musto et al., 2016b; Noia et al., 2016; Ostuni et al., 2013). Moreover, there exists little research on leveraging lightweight LOD-enabled features which can be directly queried from a SPARQL endpoint for state-of-the-art factorization approaches such as factorization machines. Our objective here is to make LOD-enabled recommendations straightforward, and reduce the additional effort when combining the background knowledge of items from a KG and user item interactions for extracting semantic features.

- (How) can we ease the process of leveraging background knowledge from KGs for item recommendations while having competitive performance compared to previous semantics-aware approaches?

In Section 6.2, we will tackle this problem by investigating different sets of lightweight LOD-enabled features such as *Predicate-Object lists*, *Subject-Predicate lists*, *PageRank scores* etc., in the context of factorization machines.

Most previous studies as well as *LODFM* exploit the existing knowledge about items in a KG for item recommendations, and therefore, do not incorporate the *incompleteness* of KGs. In addition, these studies have focused on leveraging knowledge in one direction, i.e., from KGs to the task of item recommendations. Therefore, it is not clear that whether the knowledge from item recommendations, *user-item interaction histories*, can be transferred to the KG completion task with respect to the domain of items. To answer these questions, we investigate transfer learning between the two tasks with a co-factorization model in Section 6.3.

- Does transfer learning between the two tasks improve the performance compared to the approaches without transfer learning for each task?

First, with item recommendations as the target task and KG completion as the source task, we are interested in whether incorporating the incompleteness of a KG performs better when compared to a state-of-the-art approach using a factorization machine which exploits existing knowledge from the KG, and outperforms other baselines. Second, we aim to investigate whether the knowledge can be transferred from item recommendations to KG completion and improves the performance when KG completion is the target task.

Chapter 3

Semantics-Aware User Modeling: Inferring User Interests for Active Users

Given the background knowledge on semantics-aware user modeling and recommender systems in the previous chapter, we investigate user modeling strategies for *active* users in OSNs in this chapter. The main contributions of this chapter have been published in Piao and Breslin, 2016a; Piao and Breslin, 2016d; Piao and Breslin, 2016b; Piao and Breslin, 2016c.

3.1 Introduction

With the growing popularity of OSNs, user interest profiles inferred from OSNs can be used beyond those OSNs for facilitating personalization in third-party applications. For example, the inferred user interest profiles from Twitter can be used for providing personalized recommendations in a third-party application that allows social login for users. With the continued widespread development of the social login functionality, inferring user interest profiles from their OSN activities plays a central role in many applications for providing personalized recommendations with the permission of those users, especially for cold-start users who have joined those services recently.

In this chapter, we focus on inferring user interest profiles for *active* users who continuously generate content in OSNs. A user is defined as *active user* if the user published at least 100 posts (Jain et al., 2013; Lu et al., 2012; Piao and Breslin, 2016a) in the dataset which we crawled for our experiment (some limitations of the definition of active users such as the lack of consideration for the posting distribution can be found in Section 7.2). In the next chapter, we will investigate how to infer user interest profiles for *passive* users who

have a limited number of posts but who keep following other people for receiving the information they need. We investigate four dimensions of user modeling for active users: (1) *representation* of user interest profiles, (2) *temporal dynamics* of user interests, (3) *interest propagation*, and (4) *content enrichment*. Figure 3.1 provides an overview of these four user modeling dimensions.

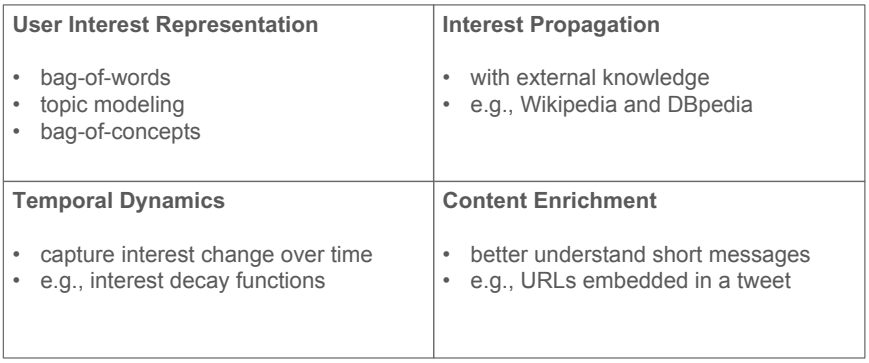


FIGURE 3.1: Four main user modeling dimensions investigated in this chapter.

In addition, we study whether these different design dimensions can be combined together to improve the quality of user interest profiles. The contributions of this chapter are summarized as follows.

- We propose various representation strategies for representing user interest profiles.
- We investigate several interest propagation strategies for entity-based user interest profiles using different aspects of DBpedia such as *classes*, *related entities* via different predicates, and *categories*.
- We compare different interest decay functions proposed in the literature, and show their comparative performance against each other, which has not been studied before.
- Finally, based on the findings through the study in each dimension, we further investigate the synergistic effect of combining those dimensions for inferring user interest profiles.

In the rest of this thesis, *entities*, *classes* and *categories* denote DBpedia *entities*, *classes* and *categories* unless otherwise noted. In addition, we use concepts to denote DBpedia entities, categories or classes. Therefore, concept-based user interest profiles can be used to denote profiles with a hybrid representation strategy leveraging DBpedia entities, categories or classes.

The rest of this chapter is organized as follows. In Section 3.2, we define user interest profiles in this thesis, and describe the evaluation strategy for inferred user interest profiles based on different user modeling strategies. In Section 3.3, we review some *category-based* user modeling strategies, and evaluate combined user interest profiles using *entities* and *categories* together. Section 3.4 investigates a weighting scheme for weighting user interests, and Section 3.5 investigates interest propagation strategies using different aspects of DBpedia beyond using *categories* only. Section 3.6 provides the comparison between different approaches for incorporating the temporal dynamics of user interests in the literature. Section 3.7 proposes a rich representation of user interest profiles which leverages DBpedia concepts as well as WordNet synsets. In Section 3.8, we study the synergistic effect of considering different user modeling dimensions together for inferring user interest profiles. Finally, Section 3.9 summarizes this chapter.

3.2 Evaluation Methodology of User Interest Profiles

In this section, we first provide the definition of user interest profiles in our approach in this thesis (Section 3.2.1), and then discuss the evaluation methodology for evaluating different user modeling strategies (Section 3.2.2). In Section 1.4.1, we give the details of a Twitter dataset which we will use for our experiments throughout this thesis.

3.2.1 User Interest Profiles

In this thesis, a user interest profile is a set of interest units (e.g., words) with their corresponding weights which denote the importance of each information unit. Formally, the generic model for representing user interest profiles is specified as follows (Piao and Breslin, 2016a).

Definition 3.2.1 (User interest profile)**.** The interest profile P_u of a user $u \in U$ is a set of weighted *interest unit* (e.g., a unit may be a DBpedia entity) where with respect to the given user u for an interest units $i \in I$ its weight is computed by a certain function $ws(\cdot)$.

$$P_u = \{(i, ws(u,i)) \mid i \in I, u \in U\} \tag{3.1}$$

Here, I and U denote the set of interest units and users, respectively. The importance of each information unit with respect to a user is determined by a weighting scheme $ws(\cdot)$.

Figure 3.2 shows a simple process for inferring user interest profiles. For example, if we use DBpedia entities for representing user interests, and use Concept Frequency (CF) as the weighting scheme $ws(u, e)$, then the weight of an entity (interest) is determined by the number of OSN activities in which user u refers to the entity e. For instance, in a Twitter profile of user u, $ws(u,$ `dbr:Google`$) = 7$ means that u the entity `dbr:Google` has been mentioned in u's tweets seven times. We further normalize user profiles so that the sum of all weights in a profile is equal to 1: $\sum_{i \in I} ws(u, i) = 1$.

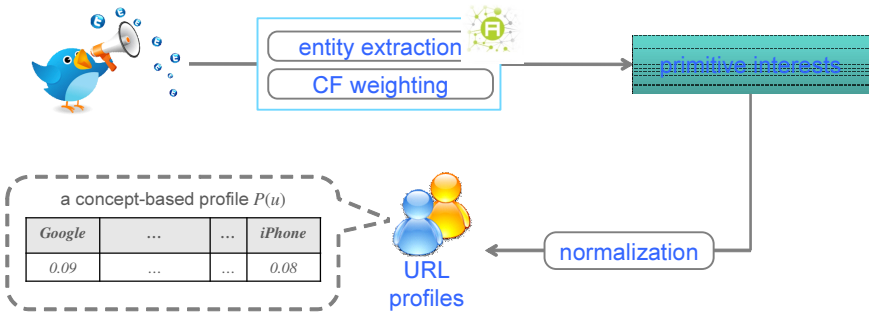


FIGURE 3.2: A simple UM process for building user interest profiles.

3.2.2 Evaluation Methodology

User modeling strategies and different user interest profiles inferred using these strategies can be evaluated in terms of an application where the inferred user interest profiles play an important role (Zarrinkalam, 2015; Abel et al., 2013b; Abel et al., 2011a) as we reviewed in Section 2.3.4. In the same way as previous studies, we evaluate different user modeling strategies in the context of a URL recommender system on Twitter where the inferred user interest profiles are used as an input.

Our main goal here is to analyze and compare the different user modeling strategies in the context of URL recommendations. We do not aim to optimize the recommendation quality, but are interested in comparing the quality achieved by the same recommendation algorithm when inputting user interest profiles based on different user modeling strategies. Therefore, we adopt a lightweight content-based algorithm as the recommendation algorithm that recommends URLs according to their *cosine* similarity with a

given user profile in the same way as previous studies (Abel et al., 2013b; Abel et al., 2011a).

Definition 3.2.2 (Recommendation algorithm)**.** Given P_u and a set of candidate URLs $L = \{P_{i1}, ..., P_{in}\}$, which are represented via profiles using the same vector representation, the recommendation algorithm ranks the candidate URLs according to their cosine similarity to the user profile (Equation 3.2).

$$cos(P_u, P_i) = \frac{P_u \cdot P_i}{\|P_u\| \|P_i\|} \tag{3.2}$$

A URL profile can be constructed by applying the same UM strategy which has been applied for building user interest profiles. Figure 3.3 shows the process of building URL profiles based on its content with the same UM strategy described in Figure 3.2.

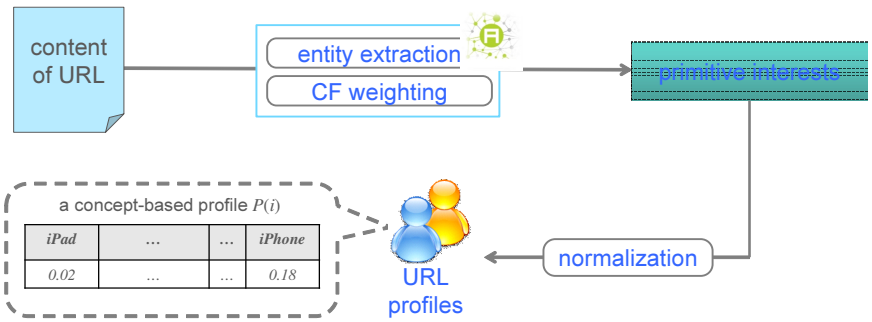


FIGURE 3.3: The process of building URL profiles with the same UM strategy described in Figure 3.2.

The ground truth of URLs, which we consider as *relevant* for a specific user, was given by the URLs shared via the user's tweets. Figure 3.4 shows an example of a ground truth URL shared by a user in a tweet.

We adopt four evaluation metrics of recommender systems, which have been used in the literature (Abel et al., 2011a; Abel et al., 2012), for evaluating the quality of URL recommendations. The four evaluation metrics are defined as follows where item denotes URL in Chapters 3-4.

- **MRR** The MRR (Mean Reciprocal Rank) indicates at which rank the first item *relevant* to the user occurs on average.

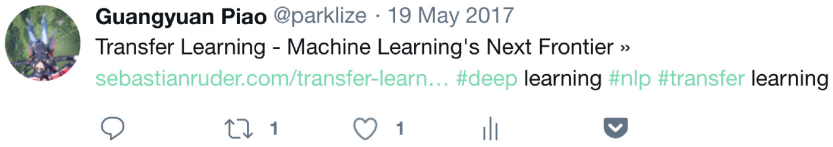


FIGURE 3.4: Example of a ground truth URL shared by a user.

$$MRR = \frac{1}{|U|}\sum_{k=1}^{|u|}\frac{1}{rank_k} \tag{3.3}$$

where U denotes the set of users, and $rank_k$ refers to the rank position where the first relevant item with respect to a user occurs.

- **S@N** The Success at rank N (S@N) stands for the mean probability that a relevant item occurs within the top-N ranked.

$$S@N = \begin{cases} 1, & if\ a\ relevant\ item\ in\ retrieved\ items\ at\ N \\ 0, & otherwise \end{cases} \tag{3.4}$$

- **R@N** The Recall at rank N (R@N) represents the mean probability that *relevant* items are successfully retrieved within the top-N recommendations.

$$R@N = \frac{|\{relevant\ items\}| \cap |\{retrieved\ items\ at\ N\}|}{|\{relevant\ items\}|} \tag{3.5}$$

- **P@N** The Precision at rank N (P@N) represents the mean probability that retrieved items within the top-N recommendations are *relevant* to the user.

$$P@N = \frac{|\{relevant\ items\}| \cap |\{retrieved\ items\ at\ N\}|}{|\{retrieved\ items\}|} \tag{3.6}$$

The *bootstrapped paired t-test*[1], which is an alternative to the paired t-test when the assumption of normality of the method is in doubt, is used for testing the significance and where the significance level is set to 0.05 unless otherwise noted.

[1] http://www.sussex.ac.uk/its/pdfs/SPSS_Bootstrapping_22.pdf

3.3 Using DBpedia Entities and Categories for Representing User Interests

Orlandi et al., 2012 proposed *category-based* user modeling strategies based on the category information for entities from DBpedia. Besides a straightforward propagation that gives equal weight to each propagated category with respect to an entity (Abel et al., 2012), Orlandi et al., 2012 also proposed a discounting strategy for those extended categories. Although *category-based* and *entity-based* user profiles showed similar performance in their user study, the authors (Orlandi et al., 2012) claimed that *category-based* user profiles produced almost seven times more user interests and these inferred interests might be helpful in the context of recommender systems. However, they did not further evaluate those user modeling strategies in the context of recommendations and left it as future work.

In this section, we discuss and evaluate two *category-based* user modeling strategies from Orlandi et al., 2012 compared to a *entity-based* one in the context of a recommender system on Twitter. We use *Tonly* to denote entity-based user interest profiles. In addition, we investigate the combined user profiles of *entity-* and *category-based* profiles, which are denoted as *Tonly+T(x)*, and evaluate them in the context of link recommendations.

3.3.1 Twitter Dataset for the Experiment

To compare and evaluate different user modeling approaches in terms of URL recommendations on Twitter, we further selected users who shared at least 10 URLs via their tweets to construct ground truth URLs from the Twitter dataset (Section 1.4.1). After all, there were 429 active users in the dataset for the experiment (41 users did not have 10 URLs in their recent posts). We used 10 URLs for each user from 429 users, as well as the URLs shared by other users but not shared by the 429 users in the dataset, for constructing candidate URLs. As a result, the set of candidate URLs consists of 5,165 distinct URLs. The rest of the tweets before the recommendation time were all used for constructing user profiles. We then adopt the evaluation strategy introduced in Section 3.2.2, which ranks URLs according to their cosine similarity scores with respect to the interest profile of a user.

3.3.2 Entity- and Category-based User Interest Profiles

Here we describe the two *category-based* user modeling methods proposed in Orlandi et al., 2012, and the combined ones of entity- and category-based

profiles for comparison. The primitive interests, i.e., DBpedia entities, are extracted using the Aylien API. Let us suppose that we have an entity-based user interest profile $Tonly = \{\ldots, (\texttt{dbr:Google}, 0.5), \ldots\}$.

- *T(Cat)* (Orlandi et al., 2012): A straightforward way of *replacing Tonly* with the categories from DBpedia, applying the same weights for the corresponding entities in the *entity-based* profiles. Given the aforementioned entity-based user interest profile, we have $T(Cat) = \{\ldots, (\texttt{dbc:Alphabet\_Inc.}, 0.5), \ldots\}$.

- *T(CatDiscount)* (Orlandi et al., 2012): Instead of the previous straightforward extension, this method applies a discounting strategy (Equation 3.7) to *T(Cat)* which discounts the weights of the propagated categories (propagated interests) from DBpedia. Therefore, given *Tonly*, $T(Cat) = \{\ldots, (\texttt{dbc:Alphabet\_Inc.}, 0.25), \ldots\}$ where the weights of propagated categories are discounted.

$$CategoryDiscount = \frac{1}{\alpha} \times \frac{1}{\log(SP + 10)} \times \frac{1}{\log(SC + 10)} \tag{3.7}$$

  where: SP = *Set of Pages belonging to the Category*, SC = *Set of Sub-Categories*. *SP* and *SC* discount the category in the context of DBpedia. Thus, a propagated category is discounted more heavily if it is a general one (i.e., the category has a great number of pages or sub-categories). In addition, we add the parameter α which denotes a discount for the propagated *category-based* user profiles when combining the *entity-based* and *category-based* user profiles. Thus, this parameter only has an effect on the combined user modeling strategies with the discounting strategy for propagated categories, i.e., *Tonly+T(CatDiscount)*. We set $\alpha = 2$ for this experiment.

- *Tonly+T(x)*: This strategy combines the *entity-based* method (i.e., *Tonly*) as well as one of the *category-based* methods mentioned above. For example, given *Tonly*, *Tonly+T(Cat)*= $\{\ldots, (\texttt{dbr:Google}, 0.5), (\texttt{dbc:Alphabet\_Inc.}, 0.5), \ldots\}$ when *T(Cat)* is used together with *Tonly*. Similarly, *Tonly+T(CatDiscount)*= $\{\ldots, (\texttt{dbr:Google}, 0.5), (\texttt{dbc:Alphabet\_Inc.}, 0.25), \ldots\}$

In the same way as Orlandi et al., 2012, a CF weighting scheme was used for the following experiment.

3.3.3 Results

Figure 3.5 and Figure 3.6 illustrate the recommendation performance of using different user modeling strategies based on category information from DBpedia as well as the performance of using *Tonly* in terms of MRR, S@N, P@N and recall.

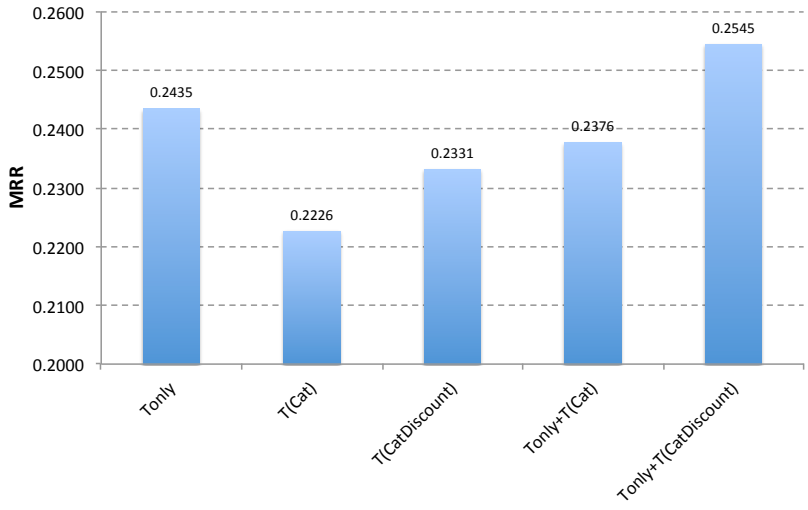
As depicted in Figure 3.5 and Figure 3.6, *Tonly+T(CatDiscount)* achieves the best performance in the context of link recommendations and significantly outperforms *Tonly* in terms of all evaluation methods. In contrast, other strategies do not perform as well as *Tonly*. For instance, *category-based* user profiles (*T(Cat)* and *T(CatDiscount)*) and the combined user profiles with the straightforward extension of categories (*Tonly+T(Cat)*) do not outperform *Tonly* but decrease the performance of link recommendations.

Different from the hypothesis from Orlandi et al., 2012, these results show that *category-based* user profiles do not perform better than *entity-based* user profiles in the context of recommender systems. However, the results indicate that the combined user profiles of *entity-* and *category-based* profiles with the discounting strategy (*Tonly+T(CatDiscount)*), improve the *entity-based* user profiles significantly, and allow the best performance in terms of link recommendations compared to other user modeling strategies.

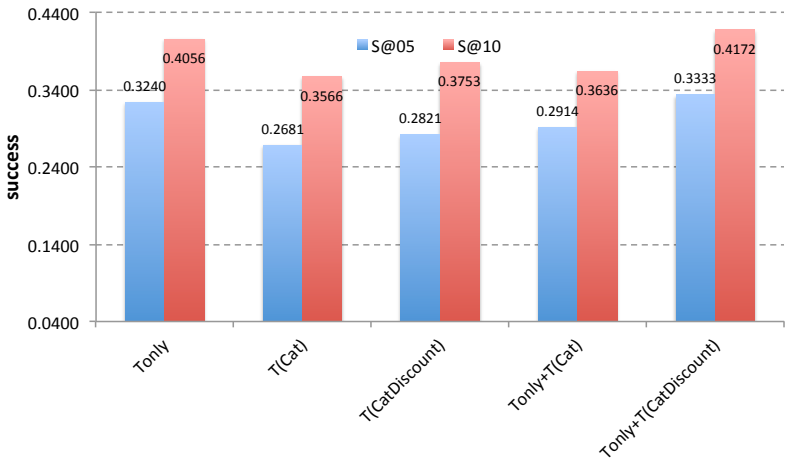
3.4 CF-IDF Weighting Scheme

In previous sections, we provided an overview of semantic user interest profiles using DBpedia entities or categories. In the following sections, we will investigate three dimensions of user modeling such as the (1) *representation* of user interest profiles, (2) *temporal dynamics* of user interests, and (3) *profile enhancement*. Before studying these dimensions, we introduce a CF-IDF weighting scheme, which we will use as our default weighting scheme afterwards instead of the CF weighting scheme.

The weighting scheme $ws(u, c)$ measures the importance of a concept with respect to a user in his/her concept-based profile. Here we use concept-based user profiles to refer to profiles represented by DBpedia entities, propagated categories, or both entities and categories. Previous studies have applied CF as the weighting scheme $ws_{CF}(u, c)$ for concept-based user profiles (Abel et al., 2011a; Orlandi et al., 2012). The weight of a concept is determined by the number of OSN activities in which a user u refers to the concept c. In contrast, we make use of CF and Inverse Document Frequency (IDF) for our weighting scheme $ws_{CF\text{-}IDF}(u, c)$, which was proposed and evaluated in the



(A) MRR

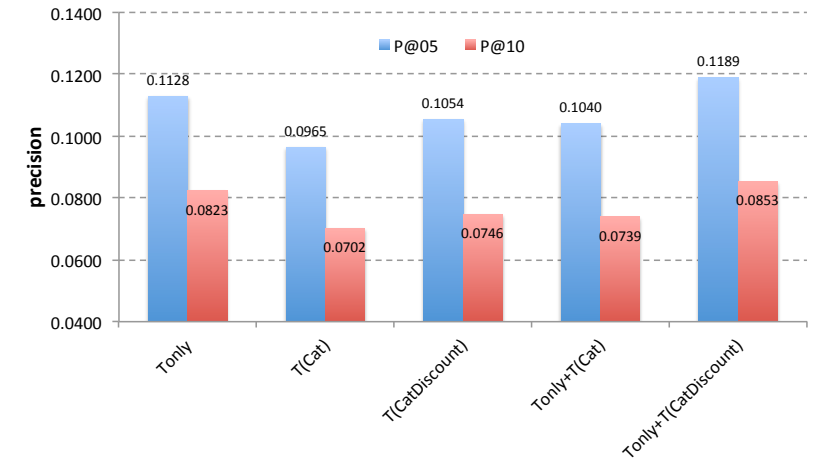


(B) Success@N

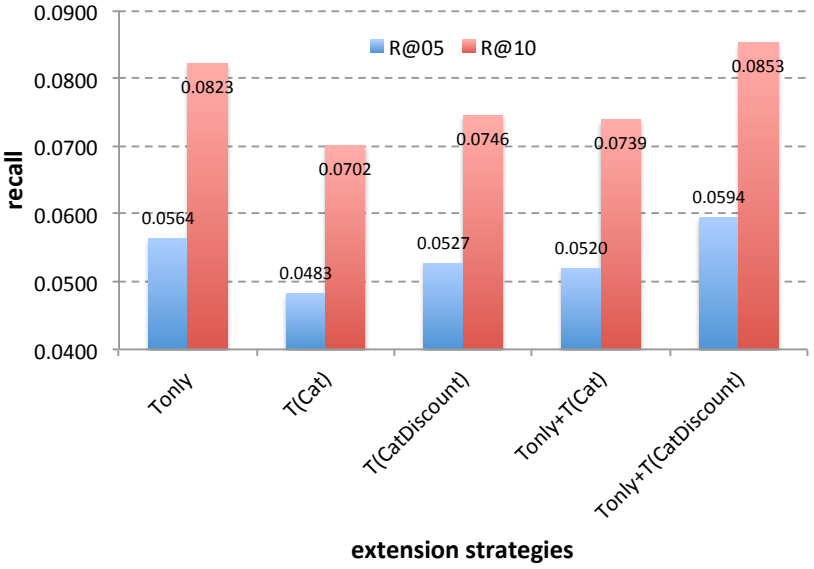
FIGURE 3.5: Performance of link recommendations in terms of MRR and S@N based on propagated user profiles using background knowledge from DBpedia

context of news recommender systems, based on a user study by Goossen et al., 2011. Similar to the TF-IDF weighting scheme used in *word-based* user modeling approaches (Abdel-Hafez and Xu, 2013), the rationale behind CF-IDF is that concepts appearing in many users' interest profiles can be discounted while concepts appearing in a specific user's profile can obtain a higher weight. More formally, it is defined as follows.

- $w_{CF}(u,c) = the\ frequency\ of\ c\ in\ a\ user's\ tweets,$



(A) Precision



(B) Recall

FIGURE 3.6: Performance of link recommendations in terms of P@N and R@N based on propagated user profiles using background knowledge from DBpedia

- $w_{CF\text{-}IDF}(u,c) = \underbrace{w_{CF}(u,c)}_{CF} \times \underbrace{\log \frac{M}{m_c}}_{IDF}$

where M is the total number of users and m_c is the number of users interested in a concept c. Figure 3.7 shows the UM process of builing user interest profiles with the CF-IDF weighting scheme on top of the one described in Figure 3.2.

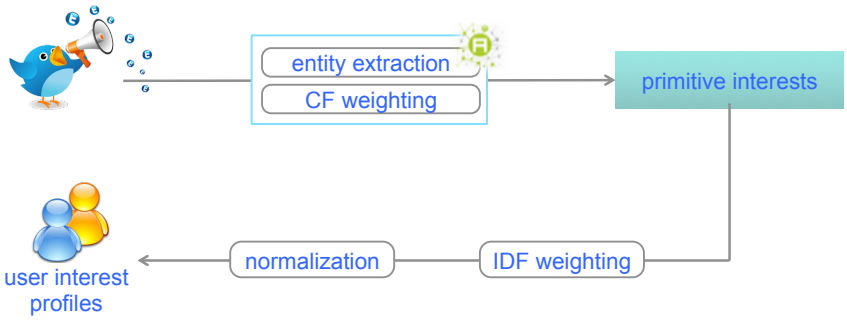


FIGURE 3.7: The UM process for building user interest profiles with the CF-IDF weighting scheme.

3.4.1 Twitter Dataset for the Experiment

We again used the Twitter dataset introduced in Section 1.4.1 for our experiment. However, in order to use the same dataset for investigating different dimensions such as the temporal dynamics of user interests in the following sections (Section 3.5-3.8), we further selected users who shared at least one URL in their tweets during the last two weeks. Also we only consider links having at least four topics (concepts) to filter out non-topical URLs (e.g., URLs sharing a person's current location via Swarm[2]). Some discussions about URLs on Twitter can be found in Section 7.2. 322 out of 480 users met the criteria who published 247,676 tweets in total.

The ground truth of links, which we consider as *relevant* for a specific user, was given by links shared via the user's tweets within the last two weeks. We used the ground truth links from 322 users, as well as the links shared by other users but not shared by the 322 users in the dataset, for constructing candidate links. In total, the ground truth of links consists of 3,959 links and the candidate set of links consists of 15,440 distinct links. The rest of the tweets before the recommendation time were all used for constructing user profiles. We adopt the evaluation strategy introduced in Section 3.2.2. For each user u, we calculate the cosine similarity between $u's$ interest profile and each candidate URL profile, and recommend ranked URLs according to their scores.

3.4.2 Comparison of CF and CF-IDF

As there was no comparison of CF and CF-IDF weighting schemes for user modeling on Twitter, we evaluated our choice of the weighting scheme in the context of link recommendations on Twitter. Figure 3.8 illustrates the recommendation performance of using CF and CF-IDF weighting schemes.

[2] https://www.swarmapp.com

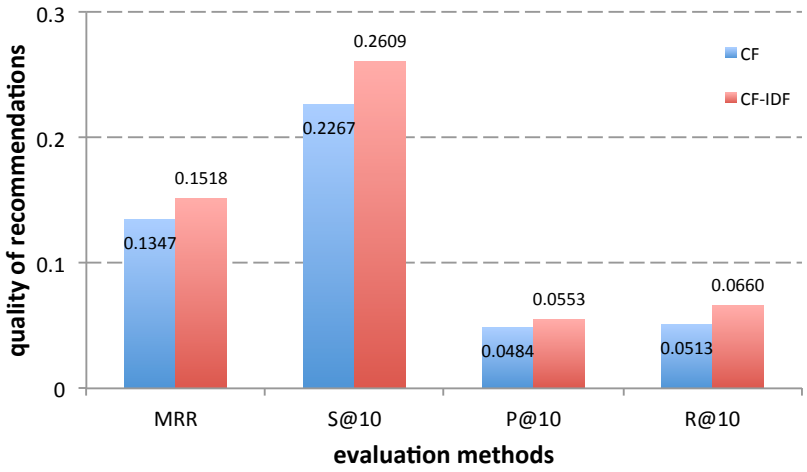


FIGURE 3.8: The quality of recommendations using CF and CF-IDF as the weighting schemes for user modeling

As we can see from the figure, the weighting scheme $w_{CF\text{-}IDF}(u,c)$ clearly outperforms the $w_{CF}(u,c)$ one in terms of all metrics and improves the recommendation performance significantly. Hence, we continue our experiments with $w_{CF\text{-}IDF}(u,c)$ as the default weighting scheme for user interest profiles.

3.5 Interest Propagation using DBpedia Graph

Previous works, either using categories only or combining entities and categories, mainly focused on a *category-based* propagation strategy using DBpedia. However, other types of information from DBpedia, i.e., *classes* (Figure 3.10 (b)) and *connected entities* via various predicates for entities from DBpedia (Figure 3.10 (c)) and the combination of them for propagating user interest profiles have not been explored.

In this section, we investigate three different types of core propagation strategies for *primitive interests*, and the combination of these core strategies. Figure 3.9 shows the UM process with interest propagation strategies. *Propagated interests* denote the interests propagated by exploring the background knowledge of DBpedia based on the extracted primitive interests.

3.5.1 Compared Core Interest Propagation Strategies

The three core interest propagation strategies based on different types of information from DBpedia are defined as follows.

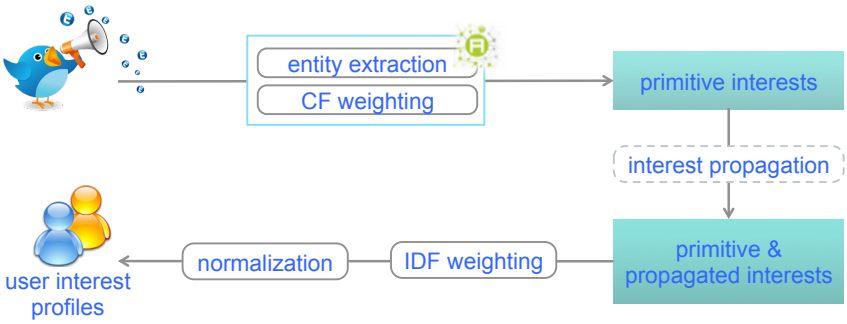


FIGURE 3.9: The UM process for building user interest profiles with interest propagation strategies.

- *Category-based*: The strategy extends *primitive interests* using their category information (Figure 3.10 (a)), which relies on the category system of Wikipedia[3] to capture the idea of a "theme", i.e., a subject of the entity (Lehmann et al., 2013).
- *Class-based*: The strategy extends *primitive interests* using their class information (Figure 3.10 (b)), which is provided via `rdf`[4]`:type` statements for all DBpedia entities using their classification from YAGO (Suchanek et al., 2007).
- *Predicate-based*: The method extends *primitive interests* with connected entities via various predicates defined in the DBpedia Ontology (Figure 3.10(c)).

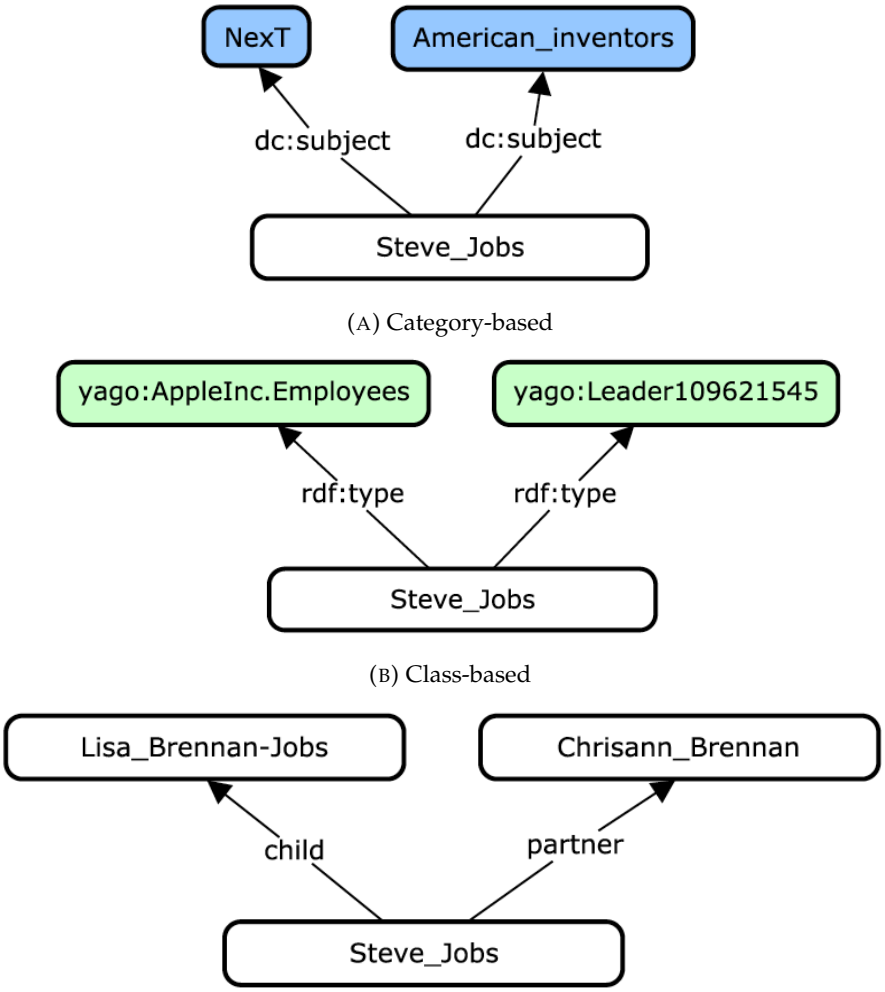
As both results from Orlandi et al., 2012 in Section 3.3.3 showed that a discounting strategy is required for the propagated concepts based on primitive interests, we adopt the same discounting strategy used in Section 3.3.2 for categories (see Equation 3.7).

In the same way as discounting the weights for propagated categories, the propagated classes using a class-based extension strategy can be discounted as follows:

$$ClassDiscount = \frac{1}{\alpha} \times \frac{1}{\log(SP' + 10)} \times \frac{1}{\log(SC' + 10)} \tag{3.8}$$

where: SP' = *Set of Pages belonging to the Class*, SC' = *Set of Sub-Classes*. The parameter α which discounts the propagation is set to 2 here as well as other core propagation strategies.

[3] `https://en.wikipedia.org/wiki/Main_Page`
[4] The prefix `rdf` denotes http://www.w3.org/1999/02/22-rdf-syntax-ns#



(A) Category-based

(B) Class-based

(C) Predicate-based

FIGURE 3.10: Three core strategies using DBpedia for extending user interests

In terms of the predicate-based extension strategy, propagated entities via different predicates are discounted based on the occurrence frequency of a specific predicate in DBpedia.

$$PredicateDiscount = \frac{1}{\alpha} \times \frac{1}{\log(P + 10)} \tag{3.9}$$

where: P = *the number of occurrences of a predicate in the whole DBpedia graph.* The intuition behind *PredicateDiscount* is that entities propagated via a predicate appearing rarely in the DBpedia graph should be given a higher weight than ones propagated via a predicate appearing frequently (Piao and Breslin, 2016f).

One of the benefits of the predicate-based extension strategy is that this

strategy strengthens the IDF value of a concept in the CF-IDF weighting scheme as the indirect mentions of the concept by users could be counted. For example, the concept `dbpedia:Montana` has appeared 36 times (which is the Document Frequency of the concept) before applying the predicate-based extension strategy. However, we observe that this number has increased to 48 after applying the extension strategy as some users indirectly mentioned the topic (e.g., `dbr:Virginia_City,_Montana` → `dbo:isPartOf` → `dbr:Montana`).

Figure 3.11 presents the number of distinct concepts in user profiles after applying the three different extension strategies. As we can see from the figure, the *category-based* extension strategy reveals more information (i.e., a greater number of concepts) in comparison to *class-* and *predicate-based* extension strategies. On average, *entity-based* user profiles have 224 concepts before any extension. After applying *category-*, *class-* and *predicate-based* extension strategies, the numbers of concepts in user interest profiles are increased to 1,865, 1,317 and 1,152, respectively. In the following, we discuss whether those user interest profiles enriched by different strategies provide better recommendation performance or not.

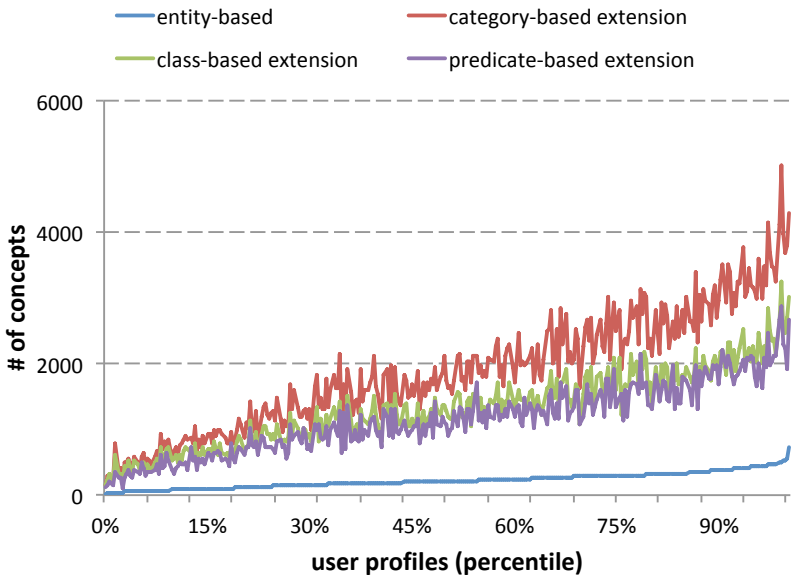


FIGURE 3.11: The number of concepts after extending user interest profiles with different core strategies

3.5.2 Results

Table 3.1 summarizes the performance of link recommendations based on user interest profiles with different extension strategies. Although there

is no significant difference between the core strategies, the *category-based* extension strategy achieves the best performance in terms of MRR while the *predicate-based* extension strategy achieves the best performance in terms of S@10, R@10 and P@10.

TABLE 3.1: The results of link recommendations based on the different strategies for extending user profiles with background knowledge from DBpedia.

extension strategy	MRR	S@10	R@10	P@10
core strategies:				
category-based	**0.2044**	0.3447	0.0928	0.0798
class-based	0.1939	0.3261	0.0861	0.0752
predicate-based	0.2017	**0.3478**	**0.0956**	**0.0804**
combined strategies:				
category & *class-based*	0.2065	0.3416	0.0914	0.0780
category & *predicate-based*	0.2083	**0.3540**	**0.0993**	**0.0820**
class & *predicate-based*	0.2063	0.3478	0.0896	0.0786
category & *class* & *predicate-based*	**0.2103**	0.3478	0.0947	0.0811

The results presented in Table 3.1 also reveal that the combination of different extension strategies for inferring user interests further enhances the quality of user modeling in the context of link recommendations. The *category* & *class* & *predicate-based* extension strategy provides the best performance in terms of MRR, and improves the performance of recommendations significantly compared to the *class-based* extension strategy. Regarding other evaluation metrics, we observe that the *category* & *predicate-based* extension strategy provides the best performance compared to other core extension strategies as well as other combined strategies. The results imply that extension strategies based on different types of information from DBpedia complement each other and the combination of these types of information can improve the quality of user modeling further.

3.6 Temporal Dynamics of User Interests

In this section, we investigate the temporal dynamics of user interests and provide a comparative study on different methods for incorporating the dynamics of user interests. To this end, we implemented various methods mentioned in Section 2.3. In the rest of this section, we describe each method in the literature in detail, and provide the evaluation results for them in

the context of link recommendations on Twitter. Figure 3.12 shows the UM process which incorporates a strategy for considering the temporal dynamics of user interests.

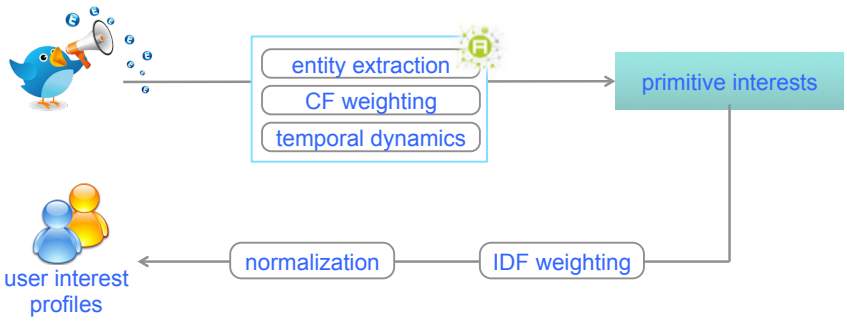


FIGURE 3.12: The UM process for building user interest profiles with a strategy for incorporating the temporal dynamics of user interests.

3.6.1 Compared Approaches

We compare the following constraint-based approaches and interest decay functions which incorporate the temporal dynamics of user interests.

- *Long-term*: *Long-term* denotes entity-based user interest profiles that are generated based on all of the historical user-generated content (UGC) of users.
- *Short-term*: *Short-term* indicates interest profiles that are generated based on the last two weeks of users' UGC before the recommendation time (Abel et al., 2011b).
- *Long-term(Orlandi)*: Orlandi et al., 2012 proposed an exponential decay function for ranking user interests as follows:

$$x(t) = e^{-t/\beta} \tag{3.10}$$

where t denotes the number of days between current time and the time of an entity mentioned in a tweet, and β is a parameter which controls the speed of the decay. In addition, they defined an initial time window (seven days) where the interests are not discounted by the decay function. We set the value of $\beta = 360 days$ in our experiment as in Orlandi et al., 2012. From a practical point of view, the interest decay function indicates that an interest value is discounted to 37% of its initial value (which is 1 by default) after 360 days.

- *Long-term(Ahmed)*: Ahmed et al., 2011 proposed getting the expected weight in terms of an interest k for user i at time t by combining three levels of abstractions using a weighted sum as below:

$$wt_{ik}^{t} = \mu_{week} wt_{ik}^{t,week} + \mu_{month} wt_{ik}^{t,month} + \mu_{all} wt_{ik}^{t,all} \tag{3.11}$$

  where $\mu_{week} = \mu$, $\mu_{month} = \mu^2$ and $\mu_{all} = \mu^3$ for $\mu \in [0,1]$. We set μ as e^{-1} in the same way as Ahmed et al., 2011 for our experiment. As this method was proposed and evaluated in terms of advertisement recommendations on web portals (i.e., Yahoo![5]), we modify μ_{week} and μ_{month} to μ_{2week} and μ_{2month} respectively to enable the method to be adapted to link recommendations on Twitter. The underlying assumption of the modification is that user interests decay slowly on Twitter as proved in a user study (Orlandi et al., 2012). We use *Long-term(Ahmedα)* to denote the modified version of *Long-term(Ahmed)*. This interest decay function combines three levels of abstractions where the decay of user interests in each abstraction is μ times the previous abstraction. In contrast, user interests in Equation 3.10 (*Long-term(Orlandi)*) decay smoothly over time.

- *Long-term(Abel)*: Abel et al., 2011a proposed a time-sensitive interest decay function, which dampens the occurrence frequency of an entity e according to the temporal distance between the entity occurrence time and the given timestamp.

$$wt(e, time, u) = \sum_{t \in T_{tweets,u,e}} (1 - \frac{|time - time(t)|}{max_{time} - min_{time}})^{d} \tag{3.12}$$

  where $T_{tweets,u,e}$ denotes the set of tweets that have been published by a user u and refer to an entity e. $time(t)$ returns the timestamp of a given tweet t and max_{time} and min_{time} denote the highest (youngest) and lowest (oldest) timestamp of a tweet in $T_{tweets,u,e}$. The parameter d is used to adjust the influence of the temporal distance. We set the parameter $d = 4$ as in Abel et al., 2011a. As we can see from Equation 3.12, this approach not only considers how old an entity e is compared to the recommendation time but also incorporates the time span of the entity in the user's historical UGC.

[5] https://yahoo.com/

3.6.2 Results

The results of the link recommendations on Twitter using the entity-based user modeling strategy with different interest decay functions are summarized in Figure 3.13. In line with the result from Abel et al., 2011b, *Short-term* profiles do not outperform *Long-term* profiles.

In terms of *Long-term(X)* user interest profiles, *Long-term(Ahmed)*, *Long-term(Ahmedα)* as well as *Long-term(Orlandi)* have comparative performance in terms of all evaluation metrics and perform significantly better than the user profiles without considering any decay of user interests (*Long-term*). *Long-term(Abel)* has slightly better performance in comparison to *Long-term* but the difference is not statistically significant ($p > 0.05$). There is a problem regarding *Long-term(Abel)* in the case of $|time - time(t)| > max_{time} - min_{time}$, i.e., $\frac{|time-time(t)|}{max_{time}-min_{time}} > 1$. In this case, we can observe that the weight is increasing with a higher value of *time*, which should be decreased instead since a higher value of *time* denotes that t is becoming older than it was before.

We also observe that *Long-term(Ahmedα)*, which slows down the decay of user interests, outperforms *Long-term(Ahmed)* consistently in terms of all evaluation methods, which shows the slow decay of user interests on Twitter. Similar conclusions were reached in Orlandi et al., 2012 based on a user study. Orlandi et al., 2012 showed in their experiment that, by setting $\beta = 360days$ in Equation 3.10 leads to better performance compared to setting $\beta = 120days$. Note that, by setting β to a larger constant in Equation 3.10 (*Long-term(Orlandi)*) as well as defining a longer period for each abstraction in Equation 3.14 (μ_{week}, μ_{month}), we are slowing down the decay of the older interests of users. The results based on different parameters of Equation 3.10 and 3.14 indicate that the quality of user modeling increases by giving a higher weight to the recent interests of users but decreases when the weight of recent interests is too high. In other words, we still need to include an older history for building user interest profiles.

3.7 Rich Representation of User Interest Profiles

Although KGs such as DBpedia provide rich semantics from background knowledge for representing and propagating user interests, they cannot cover all existing and emerging topics in OSNs. In addition, KGs lack full coverage for the lexicographic senses of lemmas, which can be provided by WordNet instead. A lemma is a word (e.g., *run*) which stands at the head of a definition in a dictionary while the word can have different forms (e.g.,

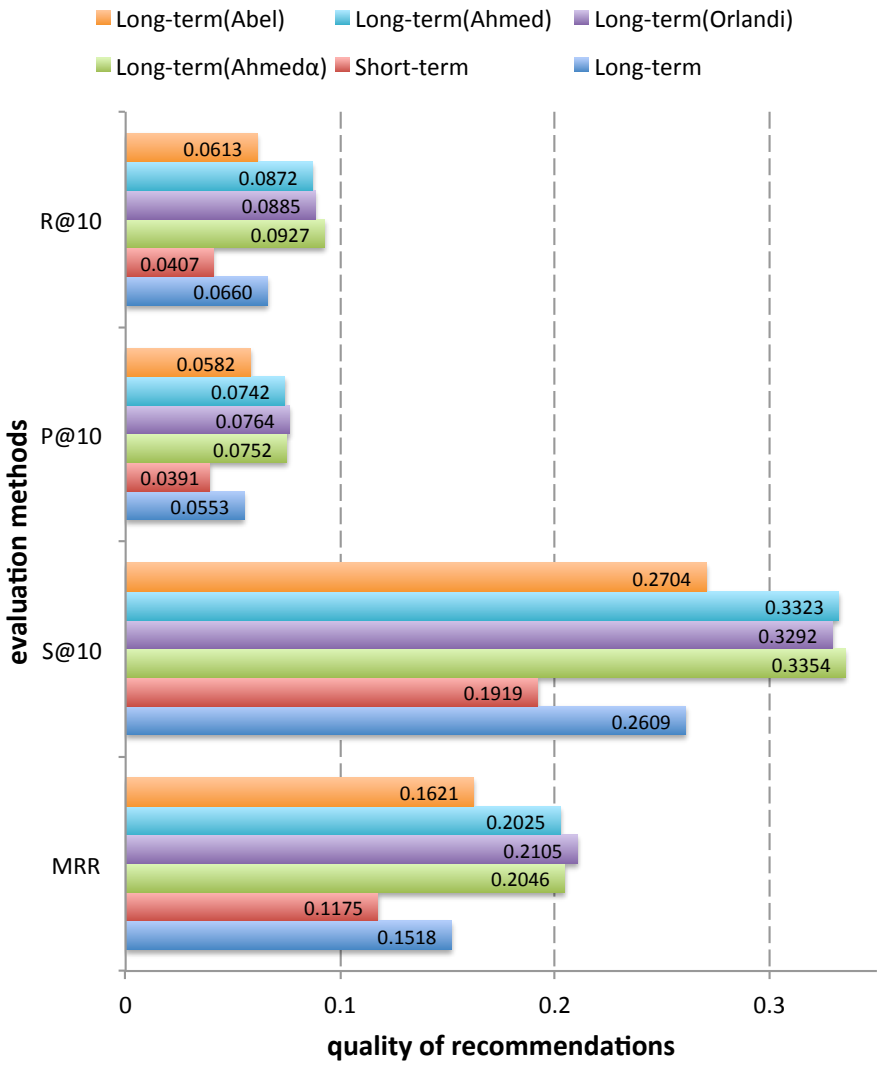


FIGURE 3.13: The quality of recommendations with different methods considering dynamics of user interests

rans, *running*, and *runs*)[6] Table 3.2 provides two real-word tweets posted by a user *@Bob* on Twitter. In the case of the second tweet posted by *@Bob*, we cannot extract any DBpedia entity from the tweet using the majority of NLP APIs mentioned in Section 1.4.2.

TABLE 3.2: Two sample tweets posted by Bob.

#1	*My Top 3 #lastfm Artists: Eagles of Death Metal(14), The Black Keys(6) & The Wombats(6)*
#2	*Just completed a 3.89 km ride. We're gonna need more...*

[6] https://simple.wikipedia.org/wiki/Lemma_(linguistics)

To circumvent this drawback, we propose using WordNet synsets and DBpedia concepts (i.e., entities or categories) together for representing user interests. Synsets in WordNet are unordered sets of synonyms - words that denote the same concept and are interchangeable in many contexts. By doing so, from the second tweet, we can extract synsets such as: s_1=[*kilometer, kilometre, km, klick (a metric unit of length equal to 1000 meters (or 0.621371 miles))*] and s_2 = [*drive, ride (a journey in a vehicle (usually an automobile))*], which denote the user interests that would be missed if a concepts-alone approach was used. Figure 3.14 shows the UM process for building user interest profiles which are represented by DBpedia concepts and WordNet synsets.

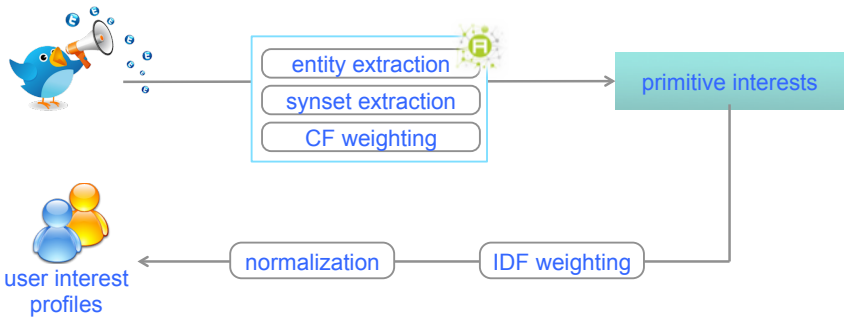


FIGURE 3.14: The UM process for building user interest profiles which are represented by DBpedia concepts and WordNet synsets.

3.7.1 Interest Extraction

As we use WordNet synsets and DBpedia concepts for representing user interests, the first step is to extract synsets and entities from the tweets of users. In the same way as in previous sections, DBpedia entities are extracted from users' tweets using the Aylien API.

To extract WordNet synsets, a WordNet-based Word Sense Disambiguation (WSD) algorithm by Degemmis et al., 2007 (see Algorithm 1), which was developed in the context of movie recommendations, has been adapted. This method extracts the WordNet synset for a word in terms of a context. In our scenario, the context of a word w is the set of words appearing in the same tweet with w (line 1) and having the same Part-Of-Speech (POS) as w. For a given tweet, our user modeling framework preprocesses with tokenization, POS tagging and lemmatization, and then uses Algorithm 1 for extracting all synsets for words based on their context. The similarity between any two synsets in Algorithm 1 (line 12) is measured as follows (Leacock and Chodorow, 1998):

$$SIMSIM(s_a, s_b) = -\log(N_p/2D) \tag{3.13}$$

where N_p is the number of nodes in the shortest path p from s_a and s_b, and D is the maximum depth of the taxonomy.

Algorithm 1: The WordNet-based WSD algorithm for tweets

input : a polysemous word w in a tweet t
output: the proper synset of w

$C \leftarrow \{w_1, \ldots, w_n\}$; // C is the context of w, i.e., other words in t with w
$X \leftarrow \{s_1, \ldots, s_k\}$; // X is the set of candidate synsets for w returned by WordNet
$s \leftarrow null$; // s is the synset to be returned
$score \leftarrow 0$; // $score$ is the similarity score assigned to s regarding the context C
$T \leftarrow \emptyset$; // T is the set of all candidate synsets for all words in C
for $w_j \in C$ **do**
    if $POS(w_j) = POS(w)$ **then**
        $X_j \leftarrow \{s_{j1}, \ldots, s_{jm}\}$;
        $T \leftarrow T \cup X_j$;
for $s_i \in X$ **do**
    for $s_h \in T$ **do**
        $score_{ih} \leftarrow SINSIM(s_i, s_h)$; // computing similarity scores between s_i and every synset $s_h \in T$
        if $score_{ih} \geq score$ **then**
            $score \leftarrow score_{ih}$;
            $s \leftarrow s_i$; // s is the synset $s_i \in X$ having the highest similarity score regarding the synsets T
return s

3.7.2 Results

To evaluate whether our new synset & concept-based user interest profiles outperform concept-based profiles, we use the entity-based user interest

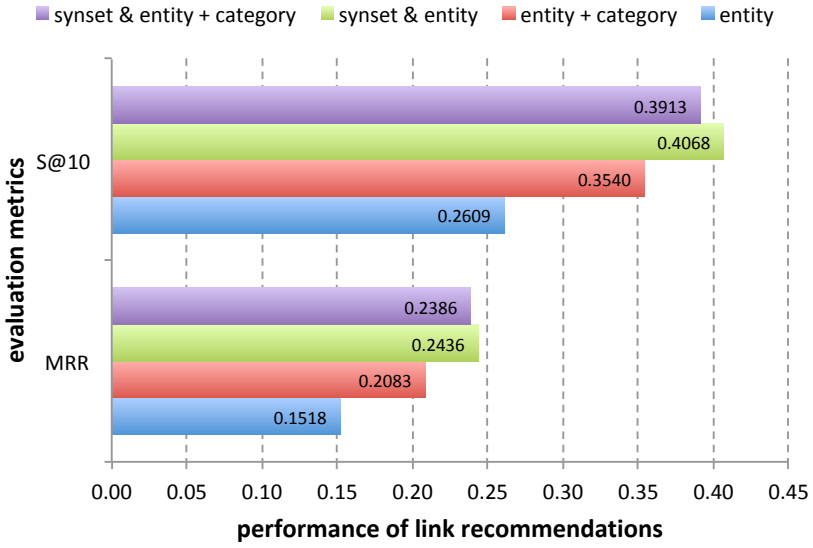
profiles (*P(entity)*) (Abel et al., 2011b) and propagated *P(entity)* using background knowledge from DBpedia (*P(entity+category)*, which is the same as *Tonly+T(CatDiscount)* in Section 3.3.2) as two baselines. The proposed approach is represented as *P(synset&entity)*, which uses synset and entities for representing user interests. In addition, the synset & entity-based user interest profiles propagated with background knowledge are denoted as *P(synset&entity+category)*.

The results of link recommendations based on different user modeling strategies in terms of the aforementioned four different evaluation metrics are presented in Figure 3.15. As we can see from the figure, there is a significant improvement for *P(synset&entity)* and *P(synset&entity+category)* compared to the concept-based approaches (*P(entity)* and *P(entity+category)*, $p < 0.05$). For example, the quality of recommendations is improved by *P(synset&entity)* by 56% and 61% in terms of S@10 and MRR, and by 77% and 87% in terms of P@10 and R@10, compared to using *P(entity)*. Similarly, using *P(synset&entity+category)* improves the recommendation performance by 11% and 15% in terms of S@10 and MRR, and by 20% and 19% in terms of P@10 and R@10 compared to using *P(entity+category)*. This indicates that using WordNet synsets and DBpedia concepts together is beneficial for user modeling on Twitter instead of using DBpedia concepts alone.

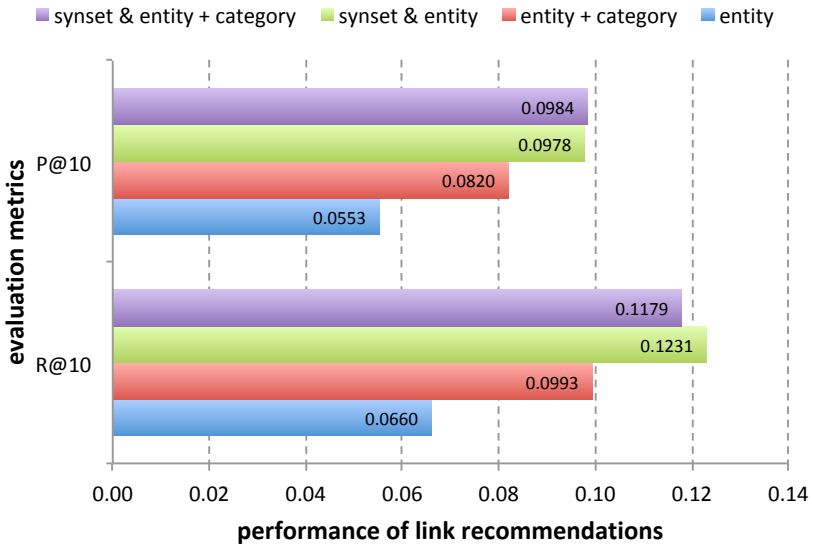
It is also interesting to observe that *P(synset&entity)*, which uses synsets and entities together without any interest propagation, has competitive performance compared to the one using the same interest representation and propagating interests with background knowledge (*P(synset&entity+category)*). This suggests that there might be little improvement by enhancing user interest profiles with a rich representation of user interest profiles. Also, it shows the importance of studying different user modeling dimensions such as *interest representation* and the *temporal dynamics* of user interests together, which has not been fully explored in the literature. Therefore, we investigate the synergistic effect of considering multiple user modeling dimensions in the next section.

3.8 A Study of Comprehensive User Modeling

In previous sections, we investigated the three dimensions: (1) *representation* of user interest profiles, (2) *temporal dynamics* of user interests, and (3) *profile enhancement* of user modeling separately. As those dimensions are not necessarily exclusive of each other, this has in turn motivated us to implement a user modeling framework which can exploit different dimensions at the same time for generating user interest profiles.



(A) MRR and success rate



(B) Precision and recall

FIGURE 3.15: Performance of link recommendations based on different user modeling strategies

In this section, we study the synergistic effect of four dimensions of user modeling including the three dimensions we discussed in previous sections. The fourth dimension is content enrichment of short microblogs. *Content Enrichment.* The ideal length of a post on any OSN ranges between 60 and 140 characters for better user engagement[7]. Therefore, there is a need to enrich this short content to better understand the context of it. Embedded URLs in a tweet can be used to enrich the short content, and provide additional information about the tweet. For example, we can follow the link in the

[7] https://goo.gl/3BoV4S

sample tweet to retrieve more information about Bob's musical interests. Many sources have shown that a large portion of tweets and retweets contain links[8,9]. We use the content in the embedded URLs of tweets to enrich short microblogs. Therefore, the entities extracted from a tweet and the content of the URL embedded in that tweet are used together for inferring user interest profiles. The Aylien API also provides the functionality for extracting entities from a given URL based on its content, which we used for our experiment for enriching short messages.

3.8.1 The Process of Generating User Interest Profiles

Figure 3.16 presents the process of generating user interest profiles for Twitter considering the aforementioned four different user modeling dimensions. The components with dotted lines are options that can be either "enabled" or "disabled" for this user modeling.

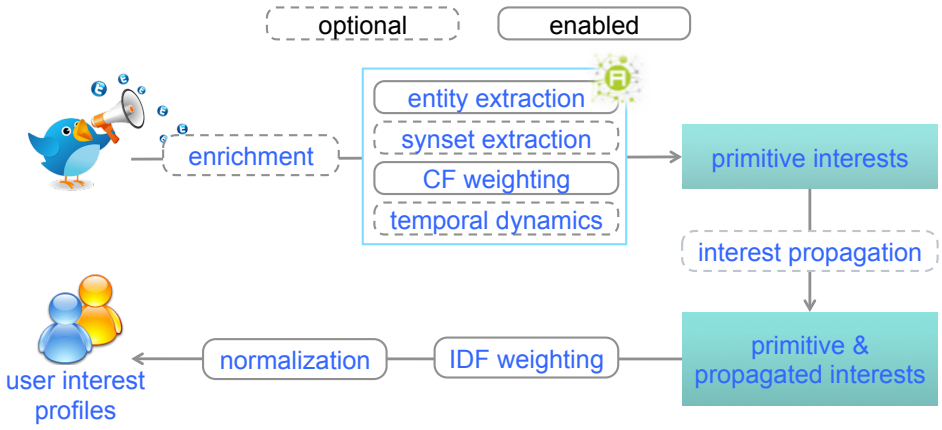


FIGURE 3.16: The process of generating user interest profiles on Twitter

As we can see from the figure, the process has three major steps:

(1) *Primitive interests extraction*. For a given user, we extract all *primitive interests* (DBpedia entities or WordNet synsets) within the UGC of a user. If the *enrichment* component is enabled, the content of links embedded in the UGC will also be used for extracting primitive interests.

- DBpedia entities are extracted using the Aylien API. For instance, the API extracts two entities `dbr:Microsoft` and `dbr:LinkedIn` from the phrase: "*Microsoft to Buy LinkedIn for $26B; LinkedIn to continue as separate brand*". Concept frequency is applied to denote the importance

[8] http://marketingrelevance.com/news/04/tweet-interesting-information/
[9] http://goo.gl/RGC16n

of a concept with respect to a user. In addition, it might adhere to strategies for incorporating the *temporal dynamics* of user interests. Stop entities like `RT_(Network)` for `@RT` in tweets are removed.

- WordNet synsets can be extracted by using Algorithm 1 at the same time as extracting entities. The rationale behinds this is that syntactic information can complement semantic information for generating user interest profiles as we have shown in Section 3.7.

(2) *Interest propagation.* This component can apply interest propagation strategies to the primitive interests of users based on background knowledge from DBpedia. The output here is a user interest profile consisting of the entities or synsets extracted from the user's tweets or the content of URLs embedded in those tweets (*primitive interests*) as well as *propagated interests* consisting of propagated categories or entities using DBpedia based on the primitive interests.

(3) *Weighting and normalization*. Finally, the user modeling framework applies Inverse Document Frequency (IDF) to the user interest profile, and further normalizes the profile so that the sum of all weights in the profile is equal to 1: $\sum_{i \in I} ws(u, i) = 1$.

Based on the optional components for user modeling (shown with dotted lines in Figure 3.16), there are 16 possible strategies which are displayed in Table 3.3. In the following subsection, we provide details of the methods for each dimension.

TABLE 3.3: The design space of user modeling, spanning 2x2x2x2=16 possible user modeling strategies.

	Interest Representation	Content Enrichment	Temporal Dynamics	Interest Propagation
Options	*DBpedia entity*	*enabled*	*enabled*	*enabled*
	synset & entity	*disabled*	*disabled*	*disabled*

3.8.2 Methods for Each Dimension

Based on the studies in each dimension in previous sections, we adopt the best strategy in each dimension for investigating their synergistic effect on user modeling.

Interest Representation: (1) DBpedia entity, or (2) WordNet synset & DBpedia entity. Entity recognition and *synsets extraction* are performed in the first step

to extract *primitive interests* from a user's tweets. We use the Aylien API for extracting DBpedia entities, and use the WSD algorithm (Algorithm 1 in Section 3.7.1) for extracting WordNet synsets.

Content Enrichment: (1) enabled, or (2) disabled. We leverage the content of links embedded in a tweet to enrich the original post content. Based on the selected option for the dimension *Interest Representation*, we apply the same extraction method for the content of embedded links. Therefore, in the case of DBpedia *entities* being used for *Interest Representation*, the *entities* extracted from the content of links embedded in tweets will also be considered as user interests if the *Content Enrichment* dimension option is enabled.

Temporal Dynamics: (1) enabled, or (2) disabled. Based on the results from the comparative study on different interest decay functions (Ahmed et al., 2011; Orlandi et al., 2012; Abel et al., 2011a) in Section 3.6, we choose a variant of the interest decay function from Ahmed et al., 2011, which performed best overall in that study. This decay function measures the expected weight in terms of an interest i for user k at time t by combining three levels of abstractions using a weighted sum as below:

$$w_{ki}^{t} = \mu_{2week} w_{ki}^{t,2week} + \mu_{2month} w_{ki}^{t,2month} + \mu_{all} w_{ki}^{t,all} \tag{3.14}$$

where $\mu_{2week} = \mu$, $\mu_{2month} = \mu^2$ and $\mu_{all} = \mu^3$ with $\mu \in [0, 1]$. We set μ as e^{-1} in the same manner as previous studies (Ahmed et al., 2011; Piao and Breslin, 2016b) for our experiment.

Interest Propagation: (1) enabled, or (2) disabled. In Section 3.5, we investigated different interest propagation strategies exploiting different types of background knowledge from DBpedia. Overall, the propagation strategy which propagates *primitive interests* with their categories (Equation 3.7) and the related entities via different predicates (Equation 3.8) in DBpedia provided the best performance compared to other state-of-the-art propagation strategies. Therefore, we use this propagation strategy for interest propagation.

3.8.3 Results

Here we present the results of experiments using different user modeling strategies in the context of link recommendations on Twitter based on the Twitter dataset (see Section 3.4.1).

In the following, let $um(representation, enrichment, dynamics, semantics)$ denote a user modeling strategy where four parameters: *representation*, *enrichment*, *dynamics* and *semantics* represent the four dimensions *Interest*

Representation, *Content Enrichment*, *Temporal Dynamics* and *Interest Propagation*, respectively. We use "disabled" to denote that a certain dimension is disabled. For instance, $um(entity, disabled, disabled, disabled)$ denotes a user modeling strategy which uses DBpedia entities for *Interest Representation* without considering any other dimensions. $um(synset \,\&\, entity, enrichment, disabled, disabled)$ denotes a user modeling strategy using synsets and entities for *Interest Representation*, and tweets are enriched by the content of embedded links when extracting user interests (i.e., the dimension *Content Enrichment* is enabled).

Table 3.4 summarizes the recommendation performance using the 16 user modeling strategies in terms of different evaluation metrics. The results are sorted in descending order in terms of MRR. Overall, the best performing strategy is $um(synset \,\&\, entity, enrichment, dynamics, disabled)$, which uses DBpedia entities and WordNet synsets for *Interest Representation*, and considers all other dimensions except *Interest Propagation*.

Another observation from Table 3.4 is the importance of (1) *Content Enrichment*, and (2) *Interest Representation* in user modeling. For instance, strategies enriching tweets with embedded links (1-8 in Table 3.4) have better performance than the ones without any enrichment (9-16), using the same option for *Interest Representation*. In terms of *Interest Representation* with or without *Content Enrichment*, we observe that using DBpedia entities with WordNet synsets (1-4 and 9-12) always provides better performance than using entities alone (5-8 and 13-16). In line with the results in Section 3.7, exploiting semantic and lexical knowledge from DBpedia as well as WordNet for *Interest Representation* improves the quality of user modeling.

Table 3.5 further illustrates statistical differences between the 16 user modeling strategies in terms of MRR. Overall, the results of other evaluation metrics are similar in terms of the MRR and thus omitted for reasons of brevity. The vertical and horizontal dimensions of the table show the comparison between the 16 strategies. As we can see from the table, there are various significant differences between the strategies ($p < 0.05$, marked in bold font). For example, strategies using entities and synsets for the dimension *Interest Representation* always significantly outperform strategies using entities, when other dimensions are kept the same (e.g., 1 and 5). The dimension *Interest Propagation* plays an important role when we use entities for *Interest Representation* without *Content Enrichment* (13-16). However, when we have a rich interest representation (i.e., using entities and synsets together) or rich content by enrichment, *Interest Propagation* has little effect on the quality of user modeling, i.e., there is no statistical difference between a user modeling strategy with *Interest Propagation* and one without any propagation (1-12). One of the possible reasons might be the rich interest

TABLE 3.4: Performance of link recommendations using 16 user modeling strategies four different evaluation metrics. The results are sorted in descending order in terms of MRR.

	User Modeling Strategies	MRR	S@10	R@10	P@10
1.	um(synset & entity, enrichment, dynamics, disabled)	0.3251	0.5062	0.1700	0.1304
2.	um(synset & entity, enrichment, dynamics, propagation)	0.3198	0.4938	0.1654	0.1298
3.	um(synset & entity, enrichment, disabled, disabled)	0.3146	0.4876	0.1595	0.1286
4.	um(synset & entity, enrichment, disabled, propagation)	0.3107	0.4752	0.1534	0.1267
5.	um(entity, enrichment, dynamics, disabled)	0.2942	0.4193	0.1405	0.1047
6.	um(entity, enrichment, disabled, disabled)	0.2886	0.4379	0.1392	0.1062
7.	um(entity, enrichment, dynamics, propagation)	0.2802	0.3975	0.1287	0.0988
8.	um(entity, enrichment, disabled,propagation)	0.2736	0.4130	0.1332	0.1006
9.	um(synset & entity, disabled, dynamics, disabled)	0.2511	0.4255	0.1257	0.0988
10.	um(synset & entity, disabled, dynamics, propagation)	0.2502	0.4193	0.1259	0.0997
11.	um(synset & entity, disabled, disabled, disabled)	0.2436	0.4068	0.1231	0.0978
12.	um(synset & entity, disabled, disabled, propagation)	0.2386	0.3913	0.1179	0.0984
13.	um(entity, disabled, disabled, propagation)	0.2083	0.3540	0.0993	0.0820
14.	um(entity, disabled, dynamics, disabled)	0.2031	0.3354	0.0927	0.0752
15.	um(entity, disabled, dynamics, propagation)	0.2024	0.3478	0.0923	0.0795
16.	um(entity, disabled, disabled, disabled)	0.1518	0.2609	0.0660	0.0553

representation, and content is giving sufficient knowledge of user interests. Additionally, the "insufficient quality" of extracted DBpedia entities from tweets using APIs, could result in inaccurate interest propagation based on the incorrect entities. This might limit the contribution of propagated interests towards user modeling.

Similar results can be found for temporal dynamics. Although considering

TABLE 3.5: Results of p-values over the 16 user modeling strategies in terms of link recommendations on Twitter (marked in bold font if $p < .05$). Strategies are sorted by MRR results as shown in Table 3.4. User modeling options are abbreviated as follows in the table: *s*: synset, *e*: entity, *en*: enrichment, *d*: dynamics, and *p*: propagation.

	seendp	seen	senp	eend	een	eendp	eenp	sed	sedp	se	sep	ep	ed	edp	e
seend	.14	.17	.11	**.01**	**.02**	**.00**	**.00**	**.00**	**.00**	**.00**	**.00**	**.00**	**.00**	**.00**	**.00**
seendp		.35	.21	**.04**	**.04**	**.01**	**.01**	**.00**	**.00**	**.00**	**.00**	**.00**	**.00**	**.00**	**.00**
seen			.24	.10	.05	**.03**	**.01**	**.00**	**.00**	**.00**	**.00**	**.00**	**.00**	**.00**	**.00**
senp				.18	.10	**.03**	**.02**	**.00**	**.00**	**.00**	**.00**	**.00**	**.00**	**.00**	**.00**
eend					.31	.05	**.03**	**.02**	**.02**	**.01**	**.01**	**.00**	**.00**	**.00**	**.00**
een						.26	.05	**.03**	**.02**	**.01**	**.01**	**.00**	**.00**	**.00**	**.00**
eendp							.26	.10	.08	.05	**.03**	**.00**	**.00**	**.00**	**.00**
eenp								.13	.13	.07	**.04**	**.00**	**.00**	**.00**	**.00**
sed									.42	.20	.08	**.01**	**.00**	**.00**	**.00**
sedp										.22	.08	**.01**	**.01**	**.00**	**.00**
se											.15	**.02**	**.01**	**.01**	**.00**
sep												**.04**	**.03**	**.02**	**.00**
ep													.32	.27	**.00**
ed														.46	**.00**
edp															**.00**
e															

Temporal Dynamics increases the performance significantly when we use entities for *Interest Representation* without *Content Enrichment* (13-16), there is no significant difference between strategies with a rich interest representation and rich content (1-12). Nevertheless, we observe that in all of the cases using entities and synsets for *Interest Representation*, considering the dimension *Temporal Dynamics* provides the best performance (see 1, 9 in Table 3.4).

To sum up, the two dimensions *Interest Representation* and *Content Enrichment* play significant roles for user modeling, followed by *Temporal Dynamics*. Although the contribution of content enrichment via embedded links might

depend on the percentage of embedded links, it is an important and valuable source for enrichment as a large number of tweets are posted with links[10]. The results also show that the *Interest Propagation* dimension has little effect on user modeling when considering different dimensions together, which is different from previous studies considering one or two dimensions (Orlandi et al., 2012; Abel et al., 2011a; Piao and Breslin, 2016b; Piao and Breslin, 2016a).

3.9 Summary

In this chapter, we investigated various user modeling dimensions for inferring user interest profiles for *active* users. We proposed interest propagation strategies based on different aspects of DBpedia beyond category-based propagation strategies. The experimental results show that the propagation strategy which explores the categories and related entities of the *primitive interests* of users provides the best performance (Section 3.5).

To better understand the comparative performance of different approaches for incorporating the temporal dynamics of user interests, we provided a comparative study using various approaches from the literature in the context of link recommendations on Twitter (Section 3.6). We also proposed a rich representation of user interest profiles that uses WordNet synsets and DBpedia concepts together in order to address the limitations of using DBpedia concepts alone (Section 3.7).

Finally, we provided a study on the synergistic effect of considering four dimensions together for user modeling, and showed that *Interest Representation* and *Content Enrichment* are the most important dimensions, followed by *Temporal Dynamics*, while *Interest Propagation* dimension has little effect on user modeling when considering all the different dimensions together (Section 3.8). The results also show the importance of studying different dimensions together when aiming towards comprehensive user modeling on microblogging services.

[10] 70% of one million tweets from the U.S. West Coast included links. http://tnw.to/s3R2i

Chapter 4

Semantics-Aware User Modeling: Inferring User Interests for Passive Users

In this chapter, we focus on user modeling strategies for *passive* users who do not have enough UGC for inferring their interest profiles as is the case for *active* users in the previous chapter. The main contributions of this chapter have been published in Piao and Breslin, 2017b; Piao and Breslin, 2017c.

4.1 Introduction

So far, we have focused on inferring user interests for *active* users who actively generate content in OSNs based on their UGC. However, the percentage of *passive users* in social networks is increasing[1], e.g., 44% of Twitter users have never sent a tweet[2] according to a research done by Twopcharts[3].

Passive users are not inactive accounts, but rather users that only consume information on social networks without generating any content. Therefore, it is important to infer the user interests of those passive users in order to provide content that they might be interested in. Firstly, recommending information that is useful for them can keep them using OSNs or may possibly make them become *active* again. Secondly, third-party applications can also utilize the inferred user interest profiles to provide personalized services for those users using social login functionality with their OSN accounts. This chapter mainly focuses on how we can infer user interest profiles for *passive users* on Twitter.

[1] http://www.corporate-eye.com/main/facebooks-growing-problem-passive-users/
[2] http://guardianlv.com/2014/04/twitter-users-are-not-tweeting/
[3] http://twopcharts.com/

Due to the fact that passive users lack of user-generated content (UGC), it is difficult to derive ground truth based on URLs shared by their UGC for evaluating the inferred user interest profiles. In this regard, we still use the Twitter users in the previous experiments but blind out all the tweets of them and use the URLs shared by the users as our ground truth. Given this setting, we use the information of their followees for inferring user interest profiles and evaluate these profiles with the ground truth. We will discuss some limitations about simulating passive user with this setting in Section 7.2.

In order to infer user interest profiles for passive users, some researchers have proposed linking the *names* of followees (those whom a user is following) to Wikipedia/DBpedia entities, and then utilizing these entities to derive abstract category-based user interests (Besel et al., 2016a). For example, if a user is following famous football players such as *@Cristiano*, they find the Wikipedia entity for `Cristiano_Ronaldo`, and then utilize the categories of the corresponding Wikipedia entity to infer user interests. Although this topical-followees approach can extract highly accurate Wikipedia entities to boost a user's interest profile, it can only link popular Twitter accounts (e.g., the accounts of celebrities) to their corresponding Wikipedia entities. As a result, the information for a large percentage of a user's followees is often ignored.

In this chapter, we investigate two types of information about followees to infer user interest profiles. First, we investigate the ***biographies (bios)*** of followees, which form an important part of followees' profiles. A ***bio*** on Twitter is a short personal description that appears in a user's profile and that serves to characterize the user's persona[4]. The length of a bio is limited to 160 characters. For example, Figure 4.1 shows a Twitter user *@bob* who has filled his bio with "*Android developer, educator.*", which describes the user's identity. The biographies of followees can be a useful information source for inferring user interest profiles for passive users compared to the topical-followees approach. For example, the Twitter account of *@UMAPconf* has a biography of "*The Conference on User Modeling, Adaptation and Personalization #umap2017*". Based on the topical-followees approach, we cannot infer any interests for a user who is following *@UMAPconf*, which has no Wikipedia entity that can be mapped to. In contrast, we can infer that the user might be

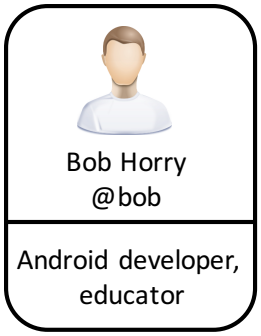


FIGURE 4.1: An example of a Twitter user profile.

[4] https://support.twitter.com/articles/166337

interested in `wiki:User_modeling` or `wiki:Personalization` based on the biography of *@UMAPconf*.

Second, we investigate a user modeling strategy that infers user interests based on the ***list memberships*** of their followees. ***List memberships*** for a user on Twitter denote topical lists which the user has been added into by the list owners. Figure 4.2 shows an example of some *list memberships* that a Twitter user *@alice* has been added to by other users on Twitter. Differring from *bios* (*self-descriptions*), *list memberships* can be seen as *others-descriptions* about *@alice*, which provide some third-party indications about what kind of topics *@alice* has been tweeting about on Twitter.

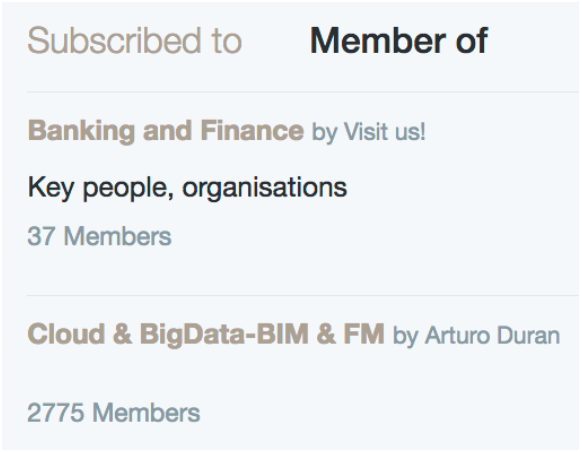


FIGURE 4.2: An example of list memberships for a Twitter user.

Finally, we explore whether the two different views (*self-descriptions* and *others-descriptions*) of followees can complement each other to improve the quality of inferred user interest profiles for *passive users* in the context of a link recommender system on Twitter.

The contributions of this chapter are summarized as follows.

- We propose user modeling strategies leveraging the *bios* of followees for interring a user's interests by investigating two different interest propagation strategies.
- We investigate whether the *list memberships* of followees can provide sufficient and qualitative information for inferring user interests for *passive users* by applying two different weighting schemes and a refined interest propagation strategy.
- We combine the two different views (*self-descriptions* and *others-descriptions*) of followees to infer user interest profiles for *passive users* in order to study the synergistic effect of combining the two views.

The rest of this chapter is organized as follows. In Section 4.2, we investigate a user modeling strategy exploring the biographies of followees to infer interest profiles for passive users. In Section 4.3, we investigate a user modeling strategy leveraging the list memberships of followees to infer user interest profiles. In addition, Section 4.4 investigates whether the two different views (*self-descriptions* and *others-descriptions*) of followees complement each other, and provide better performance in the context of link recommendations on Twitter. Finally, Section 4.5 summarizes this chapter.

4.2 Exploring the Biographies of Followees for Inferring User Interests

Before introducing our proposed approach, we first introduce two methods from the literature based on different types of information of followees for inferring user interest profiles. Afterwards, we present our proposed approach, and discuss the results compared to the two methods in the context of link recommendations on Twitter.

4.2.1 Compared Methods

SA(followees_name): Given a Twitter user u, the approach from Besel et al., 2016a leverages the names of u's followees for user modeling. The input of this approach is a Twitter account, and the output is a *category-based* user interest profile obtained via a spreading activation method. It has three main steps for generating user interest profiles.

1. Fetch a user's followees.
2. Link these to corresponding Wikipedia entities.
3. Apply a spreading activation method for the linked entities from step 2 to generate category-based profiles based on WiBi (Wikipedia Bitaxonomy[5]).

For example, if the user account *@bob* in Figure 4.1 is following *@BillGates* (the Twitter account for `Bill_Gates`), this approach searches for the name `Bill_Gates` on Wikipedia in order to find the right entity for the Twitter

[5] http://wibitaxonomy.org/

account *@BillGates* using different heuristics. We used the author's implementation[6] (Besel et al., 2016a) to link a user's followees to Wikipedia entities.

Afterwards, Wikipedia entities and categories are used as nodes in a spreading activation function. The linked Wikipedia entities are activated nodes with $ws(u,i) = 1$ for the next step where u denotes a user and i denotes a linked entity referring to one of u's followees. This approach further applies a spreading activation function from Kapanipathi et al., 2014 (see Algorithm 4.1) to propagate user interests from the extracted Wikipedia entities to Wikipedia categories, e.g., from `Bill_Gatess` to `Category:Directors_of_Microsoft`. The spreading activation function is defined as follows:

$$a_t(j) \leftarrow a_{t-1}(j) + d_{subnodes} \times b_j \times a_{t-1}(i) \tag{4.1}$$

$$d_{subnodes} = 1/\log N_{subnodes} \tag{4.2}$$

$$b_j = \frac{N_{e_j}}{N_{e_{cmax}}} \tag{4.3}$$

where j is a node (category) being activated, and i is a sub-node of j which is activating j. $d_{subnodes}$ is a decay factor based on the number of sub-nodes (sub-entities or categories) in the current category, and b_j is an *Intersect Booster* factor introduced in Kapanipathi et al., 2014. b_j is calculated by Equation 4.3, where N_{e_j} is the total number of entities activating node j, and $cmax$ is the sub-category node of j which has been activated with the maximum number of entities (Kapanipathi et al., 2014). The weight of a node is accumulated if there are several sub-nodes activating the node.

As none of the previous studies (Besel et al., 2016a; Faralli et al., 2015b) showed the performance of using followees' profiles (i.e., the names or bios of followees) compared to using followees' tweets, we also include a baseline method using the tweets of followees for inferring user interest profiles (Chen et al., 2010) to investigate the comparative performance of the two different approaches.

HIW(followees_tweet): This approach (Chen et al., 2010) extracts so-called *high-interest words* from each followee of a user u. The *high-interest words* consist of the top 20% of words in the ranked word list from a followee f's tweets. The latest 200 tweets from each followee are considered for our study, which results in over 13,940,000 tweets from the followees of 48 users (we will discuss the details of dataset later in Section 4.2.3). To construct the

[6] https://bitbucket.org/beselch/interest_twitter_acmsac16

interest profile of u, high-interest words from all followees are aggregated by excluding the words mentioned only in a single followee's tweets. Finally, the weight of each word in u's profile is measured as $ws(u,i)$ = the number of u's followees who have i as one of their high-interest words.

4.2.2 Proposed Approach

The overview of our proposed user modeling process leveraging the biographies of followees is presented in Figure 4.3, which consists of three main steps.

1. Fetch a user's followees.
2. Extract the Wikipedia/DBpedia entities referred to in the bios of followees.
3. Apply one of these interest propagation methods:
   (a) $SA(followees\_bio)$
   (b) $IP(followees\_bio)$.

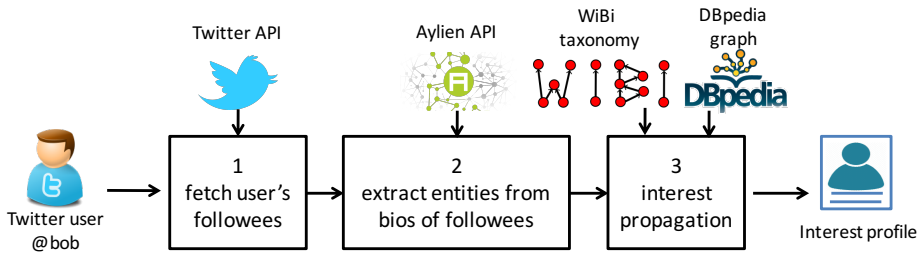


FIGURE 4.3: Overview of our proposed approach

SA(followees_bio): As one of our goals is investigating whether using the bio information of followees can improve the quality of user modeling compared to using the names of followees, we applied the same spreading activation algorithm (Algorithm 4.1) for the entities extracted from the bios of followees. Therefore, the difference between this approach and $SA(followees\_name)$ is the set of activated nodes for propagation. For $SA(followees\_bio)$, the activated nodes are extracted entities from the bios of a user's followees with $ws(u,i) = N_i$ which denotes the frequency of an interest i in their bios. Similar to $SA(followees\_name)$, the output of this approach is a *category-based* user interest profile.

IP(followees_bio): Differing from the propagation of user interests using the taxonomy of Wikipedia categories, this approach uses the *category* & *predicate-based* interest propagation introduced in Section 3.5. This propagation method extends user interests using *related entities* as well as corresponding *categories* from DBpedia. The difference between the WiBi taxonomy (Flati et al., 2014) and the DBpedia graph is presented in Figure 4.4. WiBi taxonomy can be seen as a refined hierarchical knowledge base derived from Wikipedia. As we can see from Figure 4.4 (b), the DBpedia graph provides related entities in addition to the categories of an entity. For example, as well as providing categories for the entity `Bill_Gates` via the predicate `dc`[7]`:subject`, DBpedia also gives related entities such as `Microsoft` via the predicate `dbo:board`. Therefore, as distinct from both $SA(followees\_name)$ and $SA(followees\_bio)$, the output here is a user interest profile consisting of propagated *categories* and *entities*.

4.2.3 Twitter Dataset for the Experiment

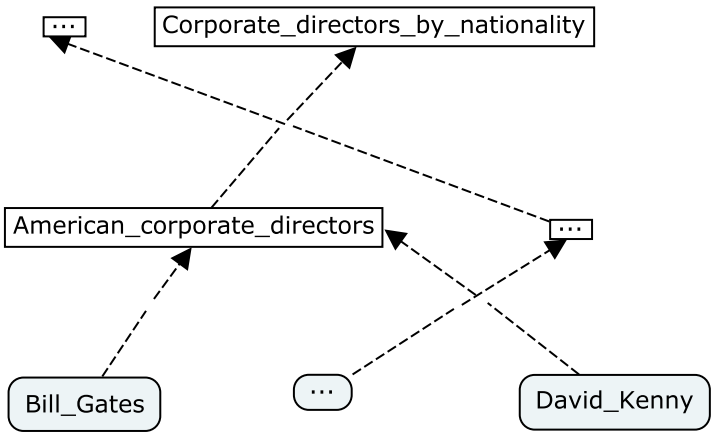
We used the Twitter dataset introduced in Section 1.4.1 for our experiment. As the focus of our study is using the followees of Twitter users for generating user interest profiles, we also crawled information on the followees for those 480 users. It was possible to crawl followees for 461 of the original 480 users via the Twitter API as some users did not exist anymore. As a result, the dataset consists of 461 users, and 902,544 followees of these users. Among these followees, we found that 812,483 users (around 90%) had filled out the bio field in their Twitter profiles. This high usage of biographies shows the potential of leveraging this information of followees for inferring user interest profiles.

As there can be a great number of followees even for a small number of users and the author's implementation (Besel et al., 2016a) to link a user's followees to Wikipedia entities requires a long time to execute for a large number of followees, we randomly selected 50 users with a corresponding set of 84,646 followees for our experiment. In the same way as previous experiments, we assumed that links shared via a user's tweets were links representing a user's interests (i.e., ground truth links), and considered links that have at least four concepts to filter out non-topical ones which were automatically generated by third-party applications such as Swarm[8]. 48 users were left as two of the 50 users had no topical links.

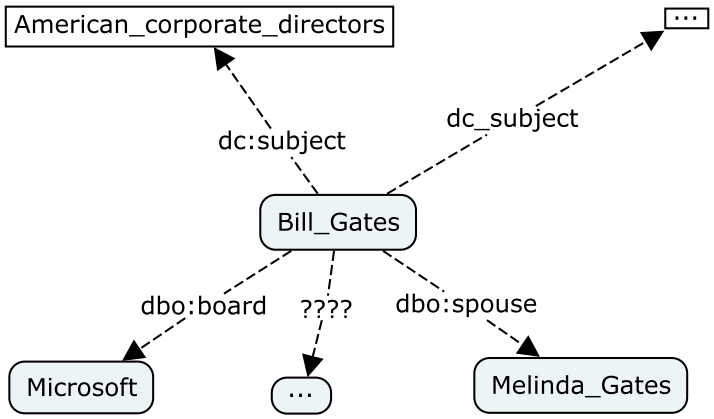
On average, there were 31.46 URLs (standard deviation: 24.5) shared by a user. The candidate set of links consists of 1,377 distinct links shared by these

[7] The prefix dc denotes `http://purl.org/dc/terms/`

[8] `https://www.swarmapp.com`



(A) WiBi taxonomy



(B) DBpedia graph

FIGURE 4.4: Examples of a WiBi taxonomy and DBpedia graph.

48 users. We then blinded the tweets of the 48 users, and used their followees' information only for building user interest profiles. The descriptive statistics of the dataset are presented in Table 4.1. These 48 users have 77,305 distinct followees in total. 10% of these followees can be linked to Wikipedia entities using the approach from Besel et al., 2016a. In contrast, 71,636 out of 77,305 (over 90%) followees have bios.

Comparison of extracted entities using names and bios. As the entities either linked via the names of followees or extracted from the bios of followees play a fundamental role in propagating user interests, we analyzed the number of entities that can be extracted using the two different sources. Figure 4.5 shows the difference between using the names and bios of followees in terms of the number of extracted entities.

TABLE 4.1: Descriptive statistics of the dataset.

# of users	48
# of followees	84,060
# of distinct followees	77,305
# of followees that can be linked to Wikipedia entities	7,694 (10%)
# of followees that have bios	71,636 (92.7%)

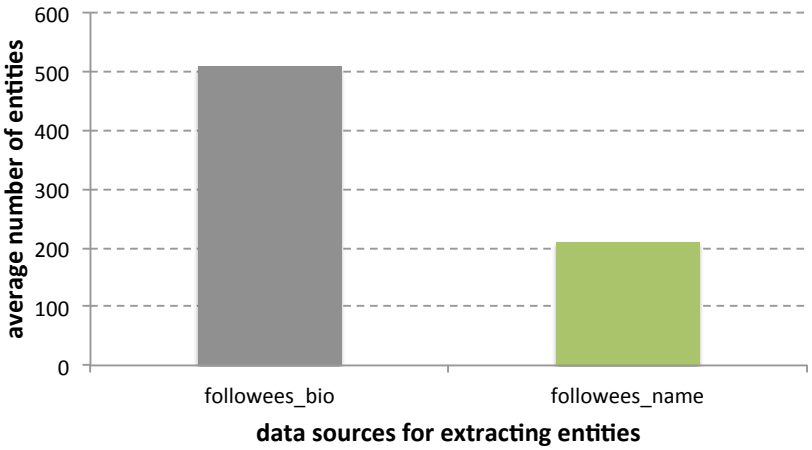


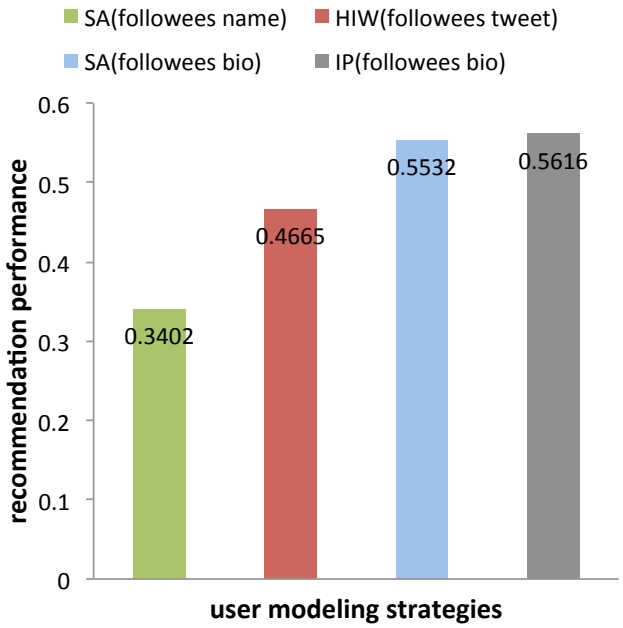
FIGURE 4.5: Number of entities extracted via names and bios of followees.

From the figure, we can observe that using the bios of followees provides more than twice the number of entities when compared to using the names of followees. On average, 509 entities can be extracted for each user using the bios of followees, and 210 entities can be extracted for each user using the names of followees. This indicates that using the bios of followees can generate more quantified user interest profiles, i.e., with a greater number of entities. We now move on to investigate whether the quantified user interest profiles generated by analyzing followees' bios have a higher quality as well, compared to those generated by linked entities based on the names of followees.

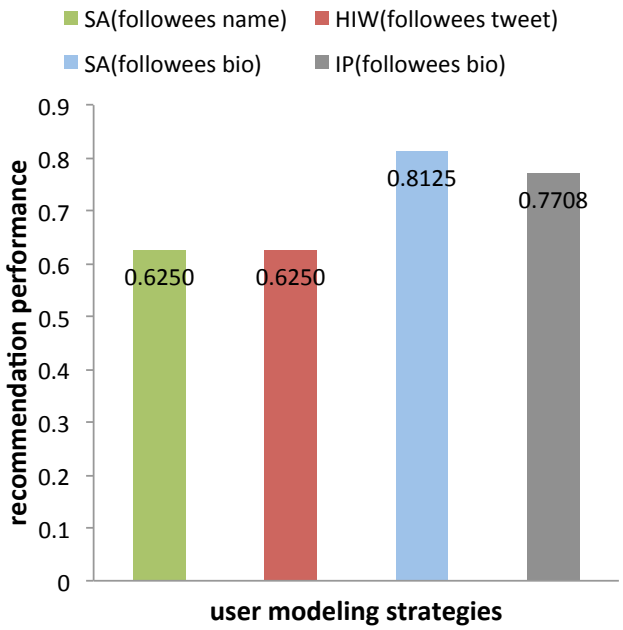
4.2.4 Results

We adopt the evaluation strategy introduced in Section 3.2.2, which ranks URLs according to their cosine similarity scores with respect to a user. Figure

4.6 and 4.7 present the results of recommendations using different user modeling strategies in terms of the four different evaluation metrics introduced in Section 3.2.2: MRR, S@10, P@10, and R@10. Overall, $IP(followees\_bio)$ provides the best performance in terms of all evaluation metrics except S@10.

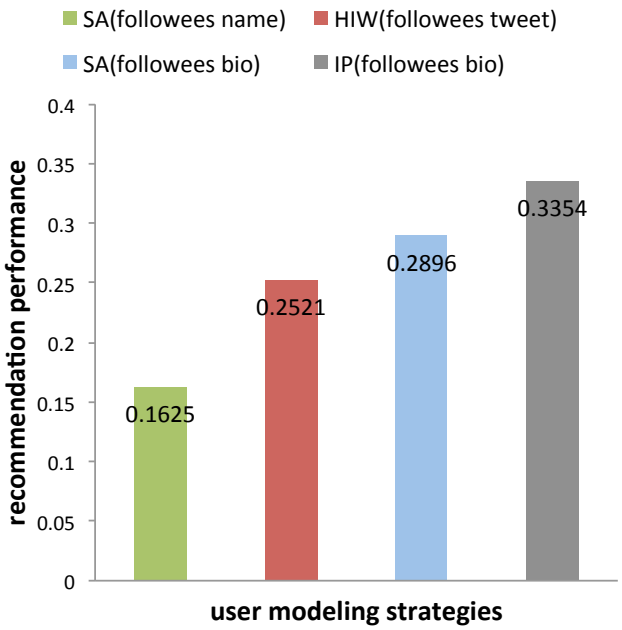


(A) MRR (Mean Reciprocal Rank)

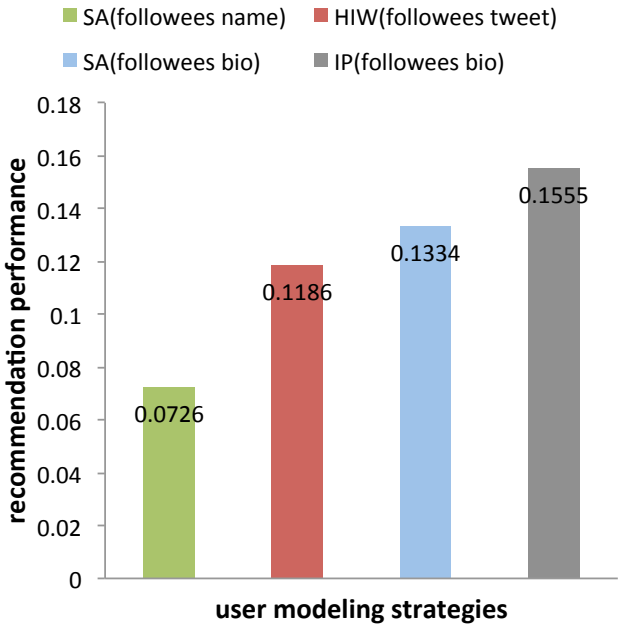


(B) S@10 (Success rate at top-10)

FIGURE 4.6: Results of the recommender system in terms of MRR an S@10.



(A) P@10



(B) R@10

FIGURE 4.7: Results of the recommender system in terms of P@10 and R@10.

Comparison between using the names and bios of followees. From Figure 4.6 and 4.7, we observe that $IP(followees\_bio)$ as well as $SA(followees\_bio)$ which use the bios of followees for user modeling outperform $SA(followees\_name)$ which uses the names of followees. A significant improvement of

$SA(followees\_bio)$ over $SA(followees\_name)$ in MRR (+63%), S@10 (+30%), P@10 (+78%), and R@10 (+84%) can be noticed ($p < 0.05$). With the same spreading activation method applied to two different sources: the names and bios of followees, the difference in terms of the four evaluation metrics clearly shows that exploring the bios of followees of passive users can infer better quality user interest profiles compared to using the names of followees in the context of link recommendations on Twitter.

Comparison between using the bios and tweets of followees. Figure 4.6 and 4.7 also show the performance of the baseline method $HIW(followees\_tweet)$, which analyzes followees' tweets for inferring *word-based* user interest profiles. The results show that our user modeling strategies using bios of followees outperform the baseline method in terms of all evaluation metrics. For instance, $IP(followees\_bio)$ outperforms $HIW(followees\_tweet)$ significantly in terms of S@10 as well as P@10 ($p < 0.05$). Considering that $HIW(followees\_tweet)$ needs to analyze over 13,940,000 tweets of followees whereas $IP(followees\_bio)$ analyzes only around 77,000 bios of followees to build interest profiles for 48 users, our approach as well as $SA(followees\_name)$ (Faralli et al., 2015b) both of which use followees' profiles (i.e., the names or bios) are more scalable in the context of OSNs such as Twitter. In contrast, the performance of $HIW(followees\_tweet)$ suggests that analyzing all the tweets of followees can lead to noisy information as an input for user modeling, which might decrease the quality of the inferred user interest profiles. For instance, a user who is following *@bob* (see Figure 4.1) might be interested in "*Android development*", however, tweets posted by *@bob* would not only contain those on the topic of "*Android development*" but also on other diverse topics that *@bob* might be interested in.

Comparison between using WiBi taxonomy and DBpedia graph. Regarding the interest propagation strategies, $IP(followees\_bio)$, which leverages the DBpedia graph for interest propagation, has better performance in terms of MRR, P@10 and R@10 when compared to $SA(followees\_bio)$. On the other hand, $SA(followees\_bio)$ has better performance in terms of S@10 than $IP(followees\_bio)$. The results suggest that $IP(followees\_bio)$ provides a greater number of preferred links to users who have successfully received recommendations, i.e., a higher P@10 value when S@10=1.

To sum up, our proposed user modeling strategy that leverages the biographies of followees provides the best performance compared to other state-of-the-art user modeling strategies, and exploring the DBpedia graph for propagating user interests has better performance compared to using the WiBi taxonomy.

4.3 Leveraging the List Memberships of Followees for Inferring User Interests

The biographies of followees provide *self-descriptions* about them. In this section, we investigate the list memberships of followees, which can be seen as *others-descriptions* about them. Figure 4.8 shows the general process of building user interest profiles based on the *list memberships* of followees. Given a Twitter user, we go through five main steps to construct an interest profile for the user.

1. Fetch all of the user's followees.
2. Fetch all *list memberships* of followees.
3. Extract DBpedia entities from the *list memberships*.
4. Construct *primitive interests* based on the extracted entities by applying a weighting scheme.
5. Apply an interest propagation strategy to *primitive interests*.

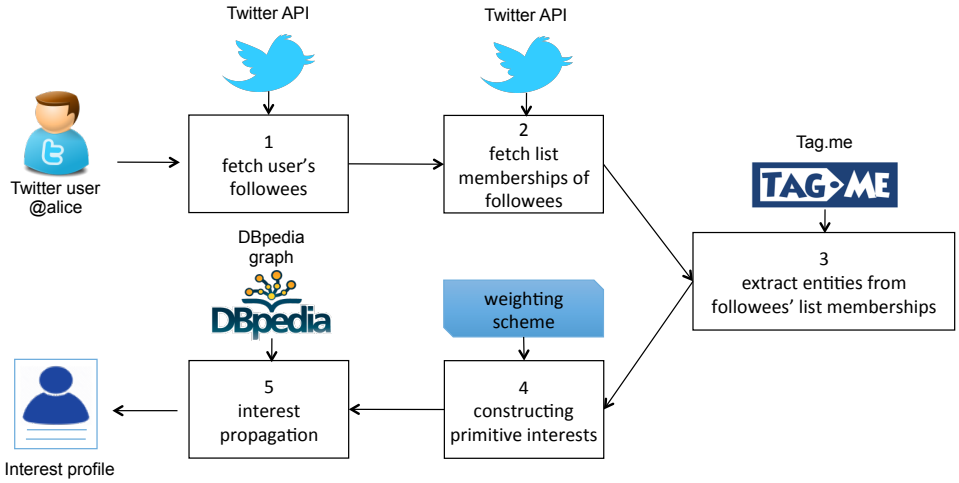


FIGURE 4.8: Overview of user modeling strategy based on followees' list memberships.

First, for a given user u, the followees of u and their *list memberships* can be fetched (steps 1 and 2) using the Twitter API. Afterwards, DBpedia entities are extracted using the tag.me API[9] based on the full names of *list memberships*. For example, entities such as `Middle_East` and `Celebrity` can be extracted from *list memberships* with the full names "*Middle East*" and "*Celebs*". Afterwards, these extracted entities are used to construct u's ***primitive interests***. Although the Aylien API has been used for extracting

[9] https://tagme.d4science.org/tagme/

entities for tweets and news articles in previous sections, we found that the Aylien API is not the optimal choice for extracting entities from the names of *list memberships* due to the short nature of those names. Therefore, we use the tag.me API instead for extracting entities from *list memberships*.

4.3.1 Constructing Primitive Interests

Those extracted entities from the list memberships of followees have to be aggregated for constructing the primitive interests of a passive user. To this end, we investigate two different weighting schemes for weighting extracted entities in order to construct a user's *primitive interests*.

- *Weighting Scheme 1 (WS1).* The intuitive way of weighting extracted entities from the *list memberships* of followees is based on the the number of occurrences of these entities. However, directly summing the number of occurrences might be biased by followees who have a large number of *list memberships*. Therefore, we use a normalized sum of occurrences of entities from followees as a weighting scheme for constructing the *primitive interests* of a target user u. For example, an interest profile of a followee $f \in F_u$ can be normalized as follows.

$$P_f = \{(c_i, ws(f, c_i)) \mid c_i \in C\} \tag{4.4}$$

  where $\sum_{c_i \in C} ws(f, c_i) = 1$. Finally, the weight of an entity c_j with respect to u is measured as below:

$$ws(u, c_j) = \sum_{f \in F_u} ws(f, c_j). \tag{4.5}$$

  where F_u denotes all the followees of a user u.

- *Weighting Scheme 2 (WS2).* For a target user u, Chen et al., 2010 aggregated the weight of each word from followees' tweets by excluding the words mentioned only in a single followee. Similarly, we aggregate the weight of each entity from followees' *list memberships* by excluding entities extracted only in a single followee. The weight of each entity in u's profile $ws(u, c_j)$ is calculated as $ws(u, c_j) =$ *the number of followees who have* c_j *in their list memberships*. Note that this weighting scheme does not care about the number of occurrences of an entity in a single followee's *list memberships*, but only counts the number of followees who have the entity in their profiles. For example, the weight of an entity c_j equals five if there are five followees of u having the entity in their *list memberships*.

4.3.2 Interest Propagation Strategy

We apply the interest propagation method introduced in Section 3.5, which extends user interests using *related entities* as well as corresponding *categories* from DBpedia. However, leveraging all DBpedia categories of entities might be noisy since many Wikipedia categories are created for Wikipedia administration. Therefore, here we also refine the DBpedia graph before applying the interest propagation method.

Extracting a subset of DBpedia categories. Similar to the approach from Kapanipathi et al., 2014, we extract a subset of all DBpedia categories which we use for our interest propagation. The subset consists of all inferred sub-categories of `dbc`[10]`:Main_topic_classifications`. However, different to Kapanipathi et al., 2014 which requires the Wikipedia dump for extracting a hierarchical category graph, we connect directly to DBpedia to extract the subset of categories by using Algorithm 2. Therefore, it can be directly extracted via the DBpedia SPARQL endpoint, and can be reproduced easily. In addition, we do not remove all administration categories (inferred sub-categories of `dbc:Wikipedia_administration`) as in Kapanipathi et al., 2014 since we found that many useful categories are in the inferred sub-categories of the administration category as well as the main topic classification, such as `dbc:Drama` (see Figure 4.9). This process results in 957,963 categories for our consideration while propagating user interests.

Algorithm 2: GetSubsetOfDBpediaCategories

procedure: getSubsetOfDBpediaCategories(topCategory)

```
1 category_dictionary = {topCategory:0}; // 0 denotes
      unprocessed
2 while size(unprocessed categories in category_dictionary) > 0 do
3     for category in unprocessed categories do
4         if category not in category_dictionary then
5             add category:0 to category_dictionary;
6 return keys of category_dictionary; // return all inferred
      sub-categories
```

Merging categories and entities with the same title. In DBpedia, many entities and categories have the same title (name), e.g., `dbr:Apple_Inc.` and `dbc:Apple_Inc.`. Considering these concepts separately as entities and

[10]The prefix `dbc` denotes `http://dbpedia.org/resource/Category:`

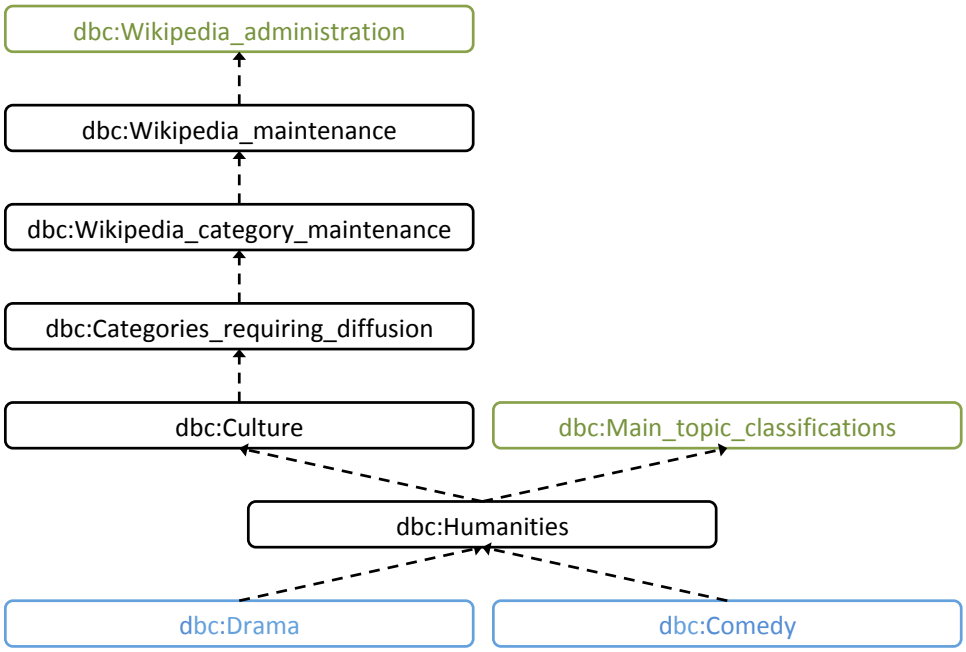
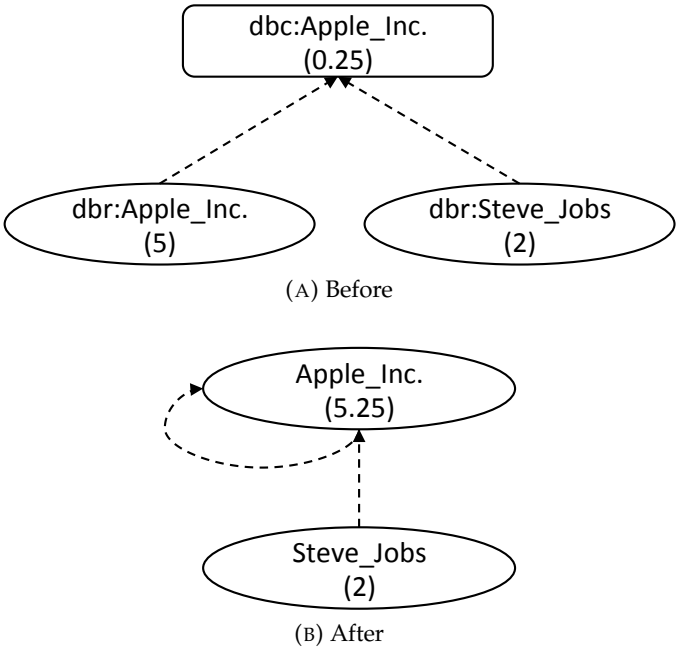


FIGURE 4.9: Example categories that belong to both `dbc:Wikipedia_administration` and `dbc:Main_topic_classifications`.

categories might decrease the quality of propagated user interests or unnecessarily increase the size of user interest profiles. Therefore, we do not treat entities and categories differently in our propagation strategy. For example, if there is a category which has the same name with an entity that has been propagated, the category and entity will be merged into a single concept, and their weights will be accumulated.

Figure 4.10 shows the difference between before merging categories and entities with the same title and after merging them. In Figure 4.10 (a), the propagated category `dbc:Apple_Inc.` has its own weight based on two entities `dbr:Apple_Inc.` and `dbr:Steve_Jobs` by considering categories and entities separately. In contrast, Figure 4.10 (b) shows that `Apple_Inc.` is considered as a single concept and its weight has been accumulated. In Section 4.3.4, we will show how trimming of categories using Algorithm 2, and the strategy merging categories and entities with the same title, positively affects the quality of inferred user interest profiles.

Finally, we apply Inverse Document Frequency (IDF) on the user interest profile P_u, and then normalize P_u in order to make the sum of all concept weights equal to 1: $\sum_{c_i \in C} ws(u, c_i) = 1$.



(A) Before

(B) After

FIGURE 4.10: Before and after merging categories and entities with the same title.

4.3.3 Twitter Dataset for the Experiment

We used the Twitter dataset introduced in Section 1.4.1, which consists of 480 randomly chosen users on Twitter with their tweets and followees. We selected 439 users who have topical URLs (URLs which have at least four entities based on their content) in their tweets from last two weeks. All of the URLs shared by each user in the last two weeks of their timelines were used to build the set of candidate URLs for recommendations. On average, each user has 2,771 followees. As the rate limits of the Twitter API for retrieving followees and *list memberships* are 15 and 75 for a 15-minute window, we only considered up to 200 followees for each user, and crawled all *list memberships* of those followees for this study. The main details of our dataset are presented in Table 4.2. Finally, the dataset corresponds to 74,488 followees for 439 users with 170 followees on average, and the candidate set of URLs consists of 15,053 distinct URLs.

TABLE 4.2: Dataset statistics.

# of passive users	**avg. # of considered followees**	**avg. # of list memberships of followees**
439	170	173

Quantitative analysis. Before discussing the performance of URL recommendations achieved using the user modeling strategy which leverages the list memberships of followees, we first look at how many list memberships a followee has been added into. The Cumulative Distribution Function (CDF) of the number of *list memberships* for 74,488 followees is shown in Figure 4.11.

The figure shows that 90% of followees have less than 492 (ln(492+1)=6.2) *list memberships*. 6,871 (9.2%) out of 74,488 followees have no *list membership*, i.e., over 90% of followees have at least one *list membership*. On average, each followee has 173 *list memberships*, which might be a useful information source of "descriptions" about a followee compared to the followee's bio. For example, 3,047 entities can be extracted from the *list memberships* of followees on average when we consider up to 50 followees for each target user in our dataset. In contrast, 23 entities can be extracted from the *bios* of followees on average. Given this quantified information from the *list memberships* of followees, we move on to investigate whether it can be leveraged for building *qualified* user interest profiles in the context of URL recommendations.

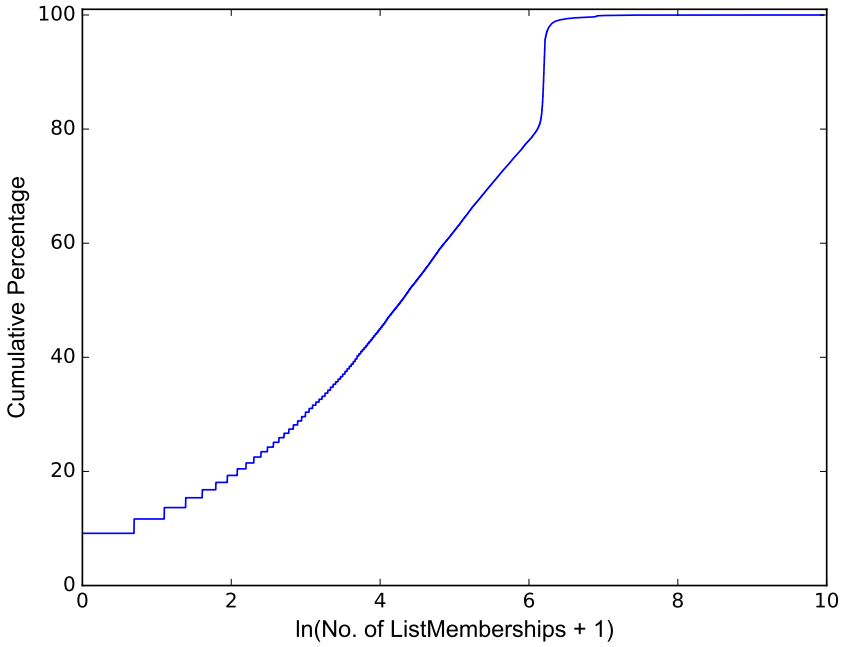


FIGURE 4.11: Cumulative distribution of the number of list memberships of followees in the dataset.

4.3.4 Comparison Between Using the List Memberships and Biographies of Followees

Table 4.3 shows the link recommendation performance using three different user modeling strategies in terms of MRR, R@10, P@10, and S@10 respectively. We also limited the number of followees for a given user from 50 to 200 in steps of 50. We use *IP(followees_bio)* presented in Section 4.2.2 as our baseline, which is denoted as $UM(followees\_bios)$ in Table 4.3.

TABLE 4.3: Recommendation performance of different user modeling strategies in terms of four different evaluation metrics and numbers of followees. The best performing user modeling strategy is in **bold**. ** denotes $p < 0.01$, and * denotes $p < 0.05$.

# of followees	Evaluation metric	$UM(followees\_bios)$ [baseline]	$UM(followees\_lm, WS1)$	$UM(followees\_lm, WS2)$
50	MRR	0.2243	**0.2622** **	0.2584 *
	R@10	0.0473	**0.0532**	0.0471
	P@10	0.1226	**0.1371** *	0.1223
	S@10	0.3690	**0.4191** *	0.4169 *
100	MRR	0.258	**0.2792**	0.2613
	R@10	0.0532	**0.0584**	0.0550
	P@10	0.1428	**0.1481**	0.1337
	S@10	0.4146	**0.4579** *	0.4442
150	MRR	0.2871	**0.2995**	0.2643
	R@10	0.0579	**0.0635**	0.0609
	P@10	**0.1535**	0.1508	0.1358
	S@10	0.4579	**0.4852**	0.4738
200	MRR	0.2952	**0.3065**	0.2638
	R@10	0.0627	**0.0653**	0.0575
	P@10	**0.1615**	0.1526	0.1353
	S@10	0.4715	**0.4920**	0.4784

Comparison between two weighting schemes in our approach. As we can see from the table, the weighting scheme $WS1$ always outperforms $WS2$ in terms of four different evaluation metrics and different numbers of followees. The result indicates that $WS1$, which applies the normalized sum of occurrences of an entity in the *list memberships* of followees, reflects the importance of the entity to passive users better when compared to the second weighting scheme which uses the number of followees having the entity in their *list memberships* ($WS2$).

Comparison between the baseline and our approach. Results in Table 4.3 also show that the user modeling strategy which exploits the *list memberships* of followees using weighting scheme 1 ($UM(followees\_lm, WS1)$) performs better than the baseline method $UM(followees\_bios)$. For example, when a passive user has less than 50 users, a significant improvement of $UM(followees\_lm, WS1)$ over $UM(followees\_bios)$ in MRR (+17%, $p < 0.01$), P@10 (+12%, $p < 0.05$), and S@10 (+14%, $p < 0.05$) can be noticed. However, we can also observe that with the number of followees of a user increasing, the difference between using $UM(followees\_lm, WS1)$ and $UM(followees\_bios)$ becomes smaller. This shows that exploiting the *list memberships* of followees can help with inferring user interest profiles in the case of a user having a small number of followees, which would be typical of "new" passive users.

Effects of DBpedia merging categories and entities with the same title. In Section 4.3.2, we introduced an interest propagation strategy by trimming DBpedia categories as well as merging categories and entities with the same title. Figure 4.12 shows the numbers of concepts (i.e., entities or categoreis) with and without merging categories and entities with the same title in terms of different numbers of followees. We found that trimming DBpedia categories as well as merging categories and entities with the same title can compress the size of user interest profiles by around 9% compared to the user modeling strategy without the trimming and merging process, while remaining at a similar performance level in the context of link recommendations.

Another observation we noticed is that the recommendation results using the *biographies* and *list memberships* of followees might complement each other. For different users, we found that using *biographies* provides better performance while using *list memberships* does not and vice versa. To test the hypothesis whether combining the two different views about followees can improve the quality of user modeling or not, we adopt an approach used in the literature for combining the two different user models in the next section.

4.4 Polyrepresentation of User Interest Profiles

The *bio* of a followee f can be seen as a *self-description* of f, while the *list memberships* of f can be seen as *others-descriptions* about f. The two different views of followees can be seen as a *polyrepresentation* of them, and the principle of *polyrepresentation* (Ingwersen, 1994) in information retrieval indicates that the overlaps between a variety of aspects or contexts with respect to a user within the information retrieval process can decrease the uncertainty and improve the performance of information retrieval. In this

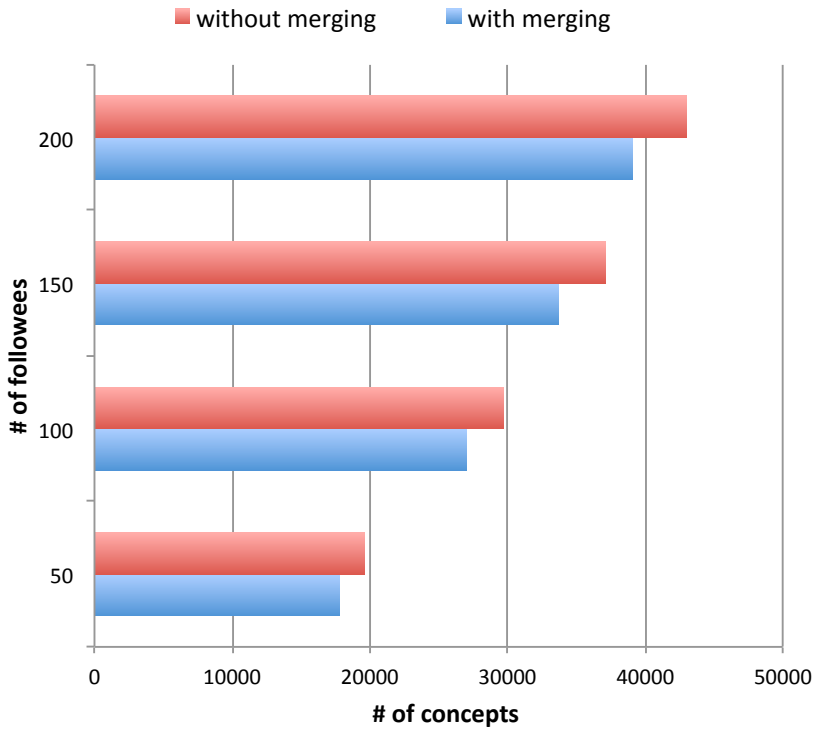


FIGURE 4.12: Number of concepts in terms of different number of followees of a user using $WS1$ with/without merging categories and entities with the same title.

section, we investigate whether combining these two different views of followees can complement each other and improves the link recommendation performance.

4.4.1 Polyrepresentation Approach

We apply a simple method used in White et al., 2009, which is based on the principle of *polyrepresentation* (Ingwersen, 1994). The approach (White et al., 2009) combined different views of a user for predicting user interests in the context of a search engine. In our context, the final rank of an item i is determined by the average rank position of each rank based on $UM(f\_bios)$ and $UM(f\_listmemberships, WS1)$:

$$final\ rank_i = \frac{1}{x_i + y_i} \tag{4.6}$$

where x_i denotes the position of i based on $UM(f\_bios)$, and y_i denotes the position of i based on $UM(f\_listmemberships, WS1)$. The higher the value is, the higher the item will be ranked. We also evaluated an alternative

approach for combining the two views which puts them into a single vector for building user interest profiles. However, the simple approach used in White et al., 2009 provides better performance than the alternative. Therefore, we report the results based on White et al., 2009 here.

4.4.2 Results

The recommendation performance of the user modeling strategy combining the two different views (*self-descriptions* and *others-descriptions*) of followees compared to the baseline user modeling strategy using bios (*self-descriptions* only) of followees is displayed in Table 4.4.

As we can see from the table, combining the two different views with a simple approach clearly outperforms the baseline method significantly in terms of four different evaluation metrics. Also, while using the *list memberships* of followees only has a significant difference compared to the baseline when the number of followees is small (i.e., # of followees = 50, 100, see Table 4.3), the combined approach has a higher significant difference ($p < 0.01$) compared to the baseline method even when the number of followees becomes larger (i.e., # of followees = 100, 150, 200, see Table 4.4).

The aforementioned combination of the two views considers the importance of each view equally (White et al., 2009). In order to investigate which view of followees has higher importance in different situations, we modify the combined score as below:

$$final\ rank_i = \frac{1}{(\beta \times x_i + (1-\beta) \times y_i)} \tag{4.7}$$

where β controls the importance of the first view, i.e., *bios* (*self-descriptions*) of followees. As one might expect, $\beta = 0$ denotes that we only consider *list memberships* (*other-descriptions*) of followees, while $\beta = 1$ denotes that we only consider *bios* (*self-descriptions*) of followees. $\beta = 0.5$ denotes that we treat two different views of followees equally as we already discussed earlier in this section (Equation 4.6).

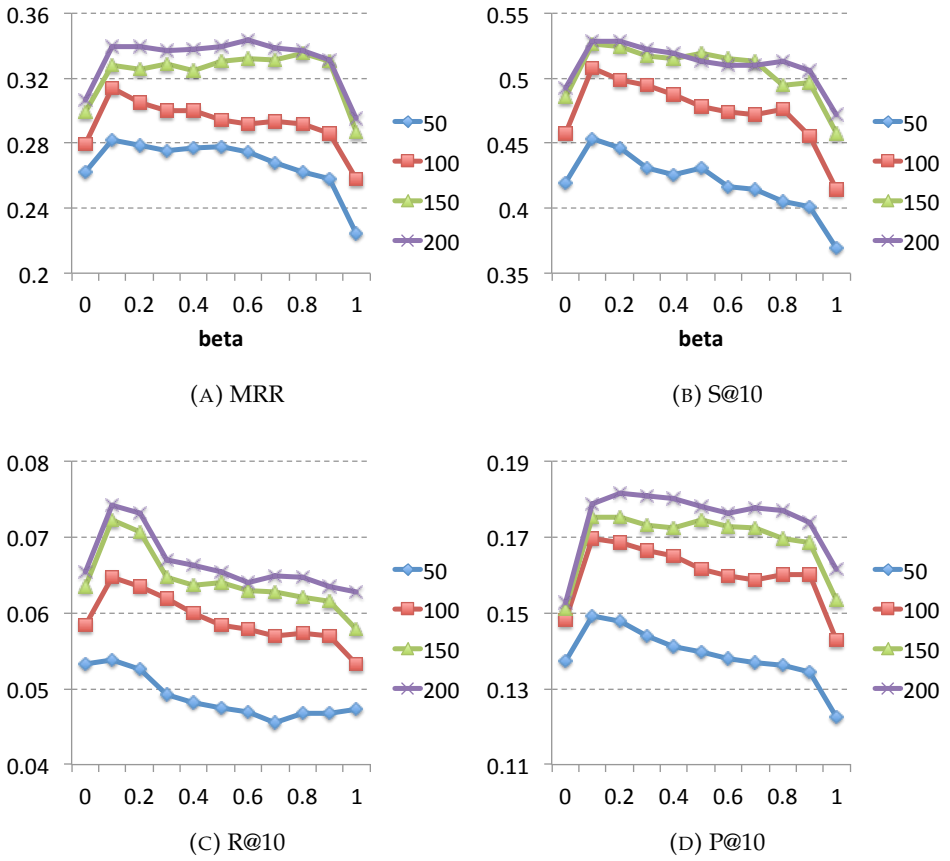
Figure 4.13 shows the link recommendation performance in terms of four evaluation metrics by setting β between 0 and 1 in steps of 0.1. As depicted in Figure 4.13, the recommendation performance is better with smaller values of β for combining the two different views (i.e., *self-descriptions* and *others-descriptions*) of followees for inferring user interest profiles in terms of R@10, P@10 and S@10. The best performance is achieved with $\beta = 0.1$, and the performance starts decreasing with increasing β. This denotes that the

TABLE 4.4: Recommendation performance of combining two views (from the bios and list memberships) of followees compared to the baseline in terms of four different evaluation metrics and numbers of followees. The best performing user modeling strategy is in **bold**. ** denotes $p < 0.01$, and * denotes $p < 0.05$.

# of followees	Evaluation metric	$UM(f\_bios)$ [baseline]	$UM(f\_bios)$ + $UM(f\_lm, WS1)$
50	MRR	0.2243	**0.2777** **
	R@10	0.0473	**0.0475**
	P@10	0.1226	**0.1396** **
	S@10	0.3690	**0.4305** **
100	MRR	0.258	**0.2946** **
	R@10	0.0532	**0.0584** *
	P@10	0.1428	**0.1615** **
	S@10	0.4146	**0.4784** **
150	MRR	0.2871	**0.3303** **
	R@10	0.0579	**0.0639** *
	P@10	0.1535	**0.1745** **
	S@10	0.4579	**0.5194** **
200	MRR	0.2952	**0.3397** **
	R@10	0.0627	**0.0654**
	P@10	0.1615	**0.1779** **
	S@10	0.4715	**0.5125** *

second view (*others-descriptions* of followees) plays a more important role for combining the two views. Similar results can be observed in terms of MRR with a small number of followees, i.e., # of followees = 50 or 100. However, as we can see from Figure 4.13 (a) that, with a big number of followees, i.e., # of followees = 150 or 200, the differences with different β values are tending towards being stable in terms of MRR.

Based on these results, we conclude that the *bios* (*self-descriptions*) and *list memberships* (*others-descriptions*) of followees can complement each other and improve the quality of user modeling in terms of link recommendations. Also, the *list memberships* of followees play a more important role for combining the two different views especially in the case of a small number of followees being available.



(A) MRR (B) S@10

(C) R@10 (D) P@10

FIGURE 4.13: The quality of user modeling with different β values for combining the two different views (*self-descriptions* and *others-descriptions*) of followees in terms of link recommendations on Twitter.

4.5 Summary

In this chapter, we proposed user modeling strategies for inferring user interest profiles for passive users based on the *biographies* (Section 4.2) and *list memberships* (Section 4.3) of their followees. The evaluation results compared to other state-of-the-art user modeling approaches indicate that both *biographies* and *list memberships* of followees provide qualitative information for inferring user interest profiles, and improve the link recommendation performance.

As bios and list memberships provide two different views about followees of users, we further investigated a polyrepresentation strategy based on these two different views (Section 4.4). Our experimental results indicate that the two different views of followees can lead to two different user interest profiles which can complement each other, and the combination of the two views provides the best performance compared to the user interest profiles

based on any single view of followees.

Although we leveraged the biographies and list memberships of followees to infer interest profiles for passive users, this information can be used for inferring interest profiles for active users as well. In Section 7.2, we discuss more about some possibilities for inferring user interests of active users leveraging the biographies and list memberships of their followees as well as the tweets of these users.

Chapter 5

Semantic Similarity Measures for LOD-enabled Recommender Systems

We have proposed user modeling strategies for inferring user interest profiles when the target users are *active* ones (see Chapter 3) or *passive* ones (see Chapter 4). Those inferred user interest profiles can be used in third-party applications which allow social login with a user's OSN profile to provide personalized recommendations, even for the first login (*cold start*) to those applications. Therefore, the focus on previous chapters has been leveraging the *implicit feedback* from users (based on their activities) in OSNs to infer their interest profiles which can be used for resolving the cold-start problem in third-party applications.

In contrast, this chapter and the next one discuss semantics-aware recommendation approaches based on *explicit feedback* from users. For example, given a user who has liked one or several Facebook pages with respect to movies, how (can) we recommend other pages matching this user's preference based on the explicit feedback (i.e., liked movie pages) and the background knowledge about those movie items from DBpedia.

This chapter deals with a *cold-start* scenario when the system has only a limited number of explicit feedback instances from users. For example, it is common that a new recommender system has a cold-start problem when users just started using the system with a small number of liked items. In this case, it is useful to recommend similar items to the item(s) that a user has liked. To this end, we propose a semantic similarity measure to calculate the similarity between two items (entities) in DBpedia, which can be used for recommending similar items sorely based on the background knowledge of items in DBpedia. The main contributions of this chapter have been published in Piao et al., 2015; Piao and Breslin, 2016f.

5.1 Introduction

Measuring similarity between entities and identifying their relatedness could be used for various applications, such as community detection in social networks or content-based recommender systems using Linked Data (Passant, 2010b). DBpedia, as a knowledge graph and a first citizen in LOD cloud, provides rich cross-domain knowledge of entities/items. In recommender systems, the similarity between items solely based on the background knowledge from a KG such as DBpedia is particularly useful in *cold-start* scenarios, e.g., providing item recommendations for a new user with limited feedback about items. In this regard, several semantic similarity/distance measures have been proposed for LOD-enabled recommendations (Passant, 2010b; Leal et al., 2012; Groues et al., 2012; Strobin and Niewiadomski, 2013). However, none of these studies evaluated against one or many of other similarity measures. Instead, each study proposed its own evaluation method for its measure. Hence, the performance compared to other similarity measures was not proven.

In this chapter, we propose a semantic similarity/distance measure *mLDSD* (modified *LDSD*) (Piao et al., 2015; Piao and Breslin, 2016f), which is built on top of a revised *LDSD*. The contributions of this chapter are summarized as follows.

- We propose a semantic similarity/distance measure on top of *LDSD* for measuring the distance between two entities in a KG.
- We evaluate our proposed method against other state-of-the-art similarity/distance measures in terms of the performance of item recommendations in the music domain.
- Finally, we investigate whether the performance of LOD-enabled recommender systems suffers from "*Linked Data sparsity*". Here, the "*Linked Data sparsity problem*" means that a lack of information on entities (e.g., small numbers of incoming/outgoing relationships from/to other entities) can decrease the performance of a recommender system.

The rest of this chapter is organized as follows. Section 5.2 provides an overview of the *LDSD* measure, and introduces the components of our proposed measure. Section 5.3 provides a preliminary evaluation of *mLDSD* using a small Last.fm dataset. In Section 5.4, we revise *mLDSD* by incorporating the number of linked resources via a link, and using a global normalization strategy. Section 5.5 describes a Facebook dataset which is larger than the Last.fm one, introduces the methods to be compared with for our evaluation, and discusses the experimental results. In Section 5.6,

we discuss the *Linked Data sparsity* problem in LOD-enabled recommender systems. Finally, we conclude this chapter with main findings in Section 5.7.

5.2 Proposed Semantic Similarity Measure

In this section, we present a similarity measure *mLDSD* (modifiend LDSD) to calculate the similarity of entities in DBpedia. This method is built on top of the *LDSD* measure, and resolves some of its limitations. In this regard, we first discuss each component of *LDSD* in Section 5.2.1, and elaborate upon their limitations. Then we describe the components of *mLDSD* in Section 5.2.2.

We use the definition of a dataset following the Linked Data principles outlined in Passant, 2010a, and the definition of a path as below:

Definition 5.2.1 (Path)**.** A dataset following Linked Data principles is a graph G such as $G = (R, L)$ in which $R = \{r_1, r_2, ...r_n\}$ is a set of entities identified by their URIs, and $L = \{l_1, l_2, ...l_n\}$ is a set of predicates identified by their URIs. A path is a sequence of entities and links between two entities, such as $p_i = \left[\ldots, \xleftarrow{l_x}, r_m, \xrightarrow{l_y}, \ldots\right]$.

For example, in the example graph (Figure 5.1), we have paths such as $[\xrightarrow{\texttt{associatedMusicArtist}}]$ and $[\xleftarrow{\texttt{musicalguests}}$, `List_of_The_Tonight_Show_with_Jay_Leno_episodes_(2013-14)`, $\xrightarrow{\texttt{musicalguests}}]$, from the entity `Ariana_Grande` to `Selena_Gomez`.

5.2.1 Linked Data Semantic Distance

Linked Data Semantic Distance (*LDSD*) (Passant, 2010b) was one of the first approaches for measuring the semantic distance between two entities on LOD datasets such as DBpedia and used for LOD-enabled recommender systems (Passant, 2010b). The distance measure (Equation 5.1) considers direct predicates from an entity r_a to r_b and vice versa. In addition, it also considers the same incoming and outgoing nodes (entities) via the same predicates of entity r_a and r_b. The distance measure has a scale from 0 to 1, where a larger value denotes less similarity between two entities. Thus, the similarity measure $LDSD_{sim}$ can be defined as Equation 5.2.

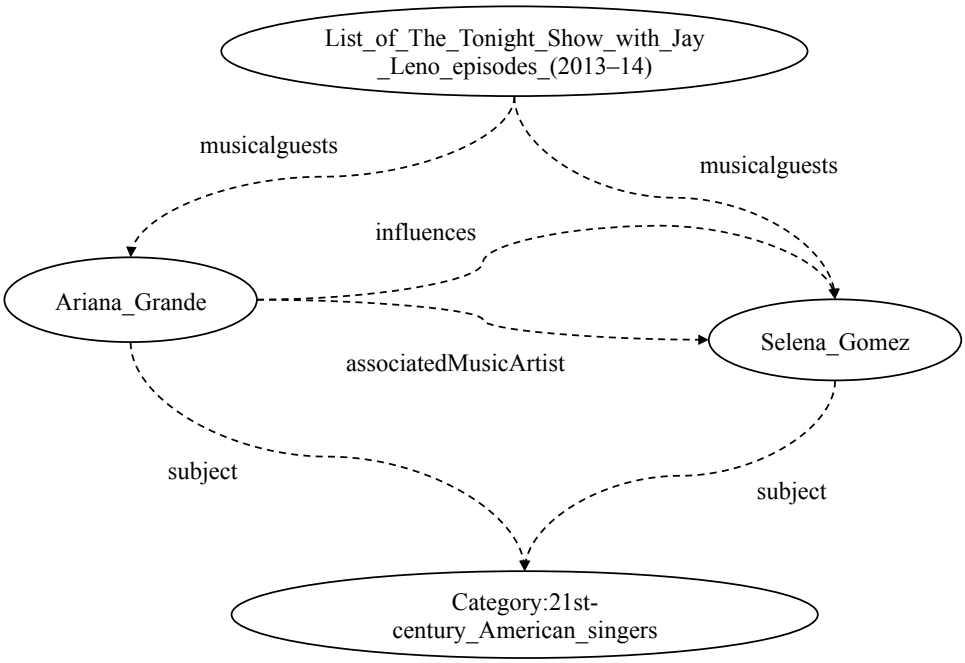


FIGURE 5.1: Example of relationships of two entities in DBpedia

$$LDSD(r_a, r_b) = \frac{1}{1 + \sum_x \frac{C_d(l_x, r_a, r_b)}{1+\log(C_d(l_x, r_a))} + \sum_x \frac{C_d(l_x, r_b, r_a)}{1+\log(C_d(l_x, r_b))} + \sum_x \frac{C_i(l_x, r_a, r_b)}{1+\log(C_i(l_x, r_a))} + \sum_x \frac{C_o(l_x, r_a, r_b)}{1+\log(C_o(l_x, r_a))}} \tag{5.1}$$

$$LDSD_{sim}(r_a, r_b) = 1 - LDSD(r_a, r_b) \tag{5.2}$$

As we can see from $LDSD$ (Equation 5.1), it consists of $C_d(\cdot)$, $C_i(\cdot)$ and $C_o(\cdot)$ functions. $C_d(\cdot)$ is a function that computes the number of direct and distinct links between entities in a graph G. For instance, $C_d(l_x, r_a, r_b)$ equals 1 if there is a link l_x from entity r_a to entity r_b. Otherwise, if there is no predicate from r_a to r_b, $C_d(l_x, r_a, r_b)$ is equal to 0. By extension $C_d(\cdot)$ can be the total number of nodes via l_x from r_a ($C_d(l_x, r_a)$). For example, in the example graph (Figure 5.1), we have: $C_d(\texttt{influences}, \texttt{Ariana\_Grande}, \texttt{Selena\_Gomez}) = 1$, $C_d(\texttt{influences}, \texttt{Ariana\_Grande}) = 1$, and $C_d(\texttt{musicalguests}, \texttt{List\_of\_The\_Tonight\_Show\_with} \ldots) = 2$.

$C_i(\cdot)$ and $C_o(\cdot)$ are functions that compute the number of indirect and distinct predicates, both incoming and outgoing, between entities in a graph G. $C_i(l_x, r_a, r_b)$ equals 1 if there is an entity r_n linked to both r_a and r_b via an incoming predicate l_x, and 0 if not. Similarly, $C_o(l_x, r_a, r_b)$ equals 1 if there is an entity r_n linked to both r_a and r_b via an outgoing predicate l_x, and 0 if not. By

extension $C_i(\cdot)$ and $C_o(\cdot)$ can be used to compute the total number of entities linked indirectly to r_a via l_i ($C_i(l_x, r_a)$ and $C_o(l_x, r_a)$). In the example (Figure 5.1), we have $C_i(\texttt{musicalguests}, \texttt{Ariana\_Grande}, \texttt{Selena\_Gomez}) = 1$ (via an incoming predicate from `List_of_The_Tonight_Show_with...`) and $C_o(\texttt{subject}, \texttt{Ariana\_Grande}, \texttt{Selena\_Gomez}) = 1$ (via an outgoing predicate to `Category:21st-century_American_singers`).

5.2.2 mLDSD Components

Equation 5.3 shows the components of our proposed semantic similarity measure for measuring the similarity between two entities in the same domain in a knowledge base.

$$mLDSD_{sim}(r_a, r_b) = \begin{cases} 1, & \text{if } URI(r_a) = URI(r_b) \text{ or } r_a \texttt{ owl:sameAs } r_b \\ 1 - mLDSD, & \text{otherwise} \end{cases} \tag{5.3}$$

Here, $mLDSD$ can be any linked data semantic distance measure which provides symmetric results, i.e., $mLDSD(r_a, r_b) = mLDSD(r_b, r_a)$. For example, Equation 5.4 shows a modified $LDSD$ (Passant, 2010b) which can be used for our $mLDSD$. Satisfying the symmetry property can reduce the calculation time for measuring the similarities between entities. For example, $mLDSD'$ only requires to measure the similarity between each pair of items. In contrast, $LDSD$ has to measure twice for each pair of items $LDSD(r_a, r_b)$ and $LDSD(r_b, r_a)$ as it does not satisfy the symmetry property.

$$mLDSD'(r_a, r_b) = \frac{1}{1 + \sum_x \frac{C_d(l_x, r_a, r_b)}{1+\log(C_d(l_x, r_a))} + \sum_x \frac{C_d(l_x, r_b, r_a)}{1+\log(C_d(l_x, r_b))} + \sum_x \frac{C_i(l_x, r_a, r_b)}{1+\log(\frac{C_i(l_x, r_a)+C_i(l_x, r_b)}{2})} + \sum_x \frac{C_o(l_x, r_a, r_b)}{1+\log(\frac{C_o(l_x, r_a)+C_o(l_x, r_b)}{2})}} \tag{5.4}$$

Table 5.1 shows the properties of $mLDSD_{sim}$, $LDSD_{sim}$, and *Shakti*. Equal self-similarity denotes that $sim(r_a, r_a) = sim(r_b, r_b)$, for all r_a and $r_b \in R$, and minimality denotes $sim(r_a, r_a) > sim(r_a, r_b)$, for all entities $r_a \neq r_b$. As the similarity calculated by $mLDSD_{sim}$ ranges from 0 to 1 and the similarity between two same items is equal to 1, $mLDSD_{sim}$ has the properties such as equal self-similarity and minimality.

TABLE 5.1: The properties of different semantic similarity/distance measures.

Property	$LDSD_{sim}$	*Shakti*	$mLDSD_{sim}$
Equal self-similarity			√
Symmetry		√	√
Minimality	√		√

5.3 Preliminary Evaluation

In Section 5.2.2, we introduced the two components of *mLDSD*, and a modified *LDSD* with the "symmetry" property. In this section, we describe a preliminary evaluation to evaluate our changes to *LDSD* so far.

5.3.1 Evaluation Metrics

We evaluate the current version of $mLDSD_{sim}$ compared to $LDSD_{sim}$ and *Shakti* in terms of recommending similar items in the music domain based on background knowledge about items from DBpedia. The recommender system recommends the top-N similar music artists for a given music artist based on the similarities between all candidates and the music artist.

The performance of the recommendations was measured by means of MRR, P@N and R@N which are defined in Section 3.2.2, and nDCG@N (Normalized Discounted Cumulative Gain). For item recommendations with respect to a set of users U, these evaluation metrics can be defined as below:

- **nDCG@N**: Precision and recall consider the relevance of items only. In contrast, nDCG takes into account the relevance of items as well as their rank positions.

$$nDCG@N = \frac{1}{IDCG@N} \sum_{k=1}^{N} \frac{2^{\hat{r}_{uk}} - 1}{\log_2(1+k)} \tag{5.5}$$

We use N = 1, 5 and 10 in the evaluation, and report the results of averaged nDCG@N, P@N and R@N over the 10 randomly selected entities of `dbo:MusicArtist` or `dbo:Band` based on different semantic similarity/distance methods. For example, if a user is interested in the music artist `dbr:Ariana_Grande`, the candidate list consists of the top 10 similar music artists recommended by Last.fm (that can be found in DBpedia) and 200 randomly selected entities of type `dbo:MusicArtist` or `dbo:Band`. Then we calculate the similarities between `dbr:Ariana_Grande` and the candidate list

with similarity measures to get the top-N recommendations. Our goal is to see the performance of the top-N recommendations based on different similarity/distance measures.

5.3.2 Last.fm Dataset

In Passant, 2010a, the authors evaluated the *LDSD* measure in the music domain by comparing against a list of recommendations from Last.fm. Last.fm offers a ranked list of similar artists/bands for each artist/band based on their similarities. They showed that in spite of a slight advantage for Last.fm, *LDSD* based recommendations achieved a reasonable score, especially considering that it does not use any collaborative filtering approach, and relies only on links between items/entities in the DBpedia graph.

Similarly, we adopt the list of recommendations from Last.fm to evaluate the performance of our recommendations. First of all, all entities of type `dbo:MusicArtist` or `dbo:Band` were extracted via the DBpedia SPARQL endpoint. By doing so, 75,682 entities were obtained consisting of 45,104 entities of type `dbo:MusicArtist`, and 30,578 entities of type `dbo:Band`. Then we randomly selected 10 entities out of these 75,682 entities. For each entity (a music artist or band in this case), we manually get the top 10 recommendations from Last.fm which can be found in DBpedia. To construct a candidate list for recommendations, we created a candidate list with these top 10 recommendations from Last.fm and 200 randomly selected entities among the 75,682 entities of type `dbo:MusicArtist` or `dbo:Band`.

5.3.3 Compared Methods

We evaluate the current version of *mLDSD* compared to $LDSD_{sim}$ and *Shakti* in terms of recommending similar items in the music domain based on items' background knowledge from DBpedia. As DBpedia provides a large set of predicates for each item, it is necessary to select a subset of domain-dependent predicates (Musto et al., 2014; Di Noia et al., 2012b).

For the *Shakti* similarity measure, we use the weights of predicates manually assigned by the authors in Leal et al., 2012. In *Shakti*, seven predicates related to the music domain were considered such as `dbo:genre`, `dbo:instrument`, `dbo:influences`, `dbo:associatedMusicalArtist`, `dbo:associatedBand`, `dbo:currentMember` and `dbo:pastMember`. Since the *Shakti* similarity measure uses the value of max step for the extension of the paths between two entities, we use 3 and 5 for the value of max step, and denote these variants as *Shakti3* (max step set to 3) and *Shakti5* (max step set to 5).

For $LDSD_{sim}$ and $mLDSD_{sim}$, we selected 15 predicates related to the music domain (see Table 5.2) for measuring the semantic similarity between entities. `dct`[1]`:subject` relates an entity to its categories. In addition, we decided to leverage the predicates belonging to the DBpedia Ontology since they represent high-quality, clean and well-structured information (Ostuni et al., 2013).

TABLE 5.2: Predicates selected for the music domain for semantic similarity/distance measures.

`dct:subject, dbo:genre, dbo:associatedBand,`
`dbo:associatedMusicalArtist, dbo:instrument,`
`dbo:formerBandMember, dbo:currentMember,`
`dbo:influencedBy, dbo:pastMember, dbo:bandMember`
`dbo:associatedAct, dbo:influenced, dbo:hometown`
`dbo:recordLabel, dbo:occupation`

5.3.4 Results

The results of recommendations for the randomly selected 10 music artists/bands are displayed in Figure 5.2. $LDSD_{sim}$ and current $mLDSD_{sim}$ have similar performance in terms of all evaluation metrics, and perform significantly better than *Shakti3* and *Shakti5* which have 3 and 5 as their max steps.

To summarize, the current version of $mLDSD_{sim}$ has similar performance for measuring the similarities of entities compared to $LDSD_{sim}$. In the next section, we further investigate some limitations of $mLDSD_{sim}$, and modify and evaluate it in terms of recommender systems.

5.4 Modified Distance Measures for mLDSD

In this section, we propose two semantic distance measures on top of $mLDSD'$ (Equation 5.4). The first one incorporates the number of linked entities via a predicate, and the second one uses a global normalization strategy instead of local normalization based on the local context of two entities.

[1]The prefix `dct` is used for the namespace `http://purl.org/dc/terms/subject`

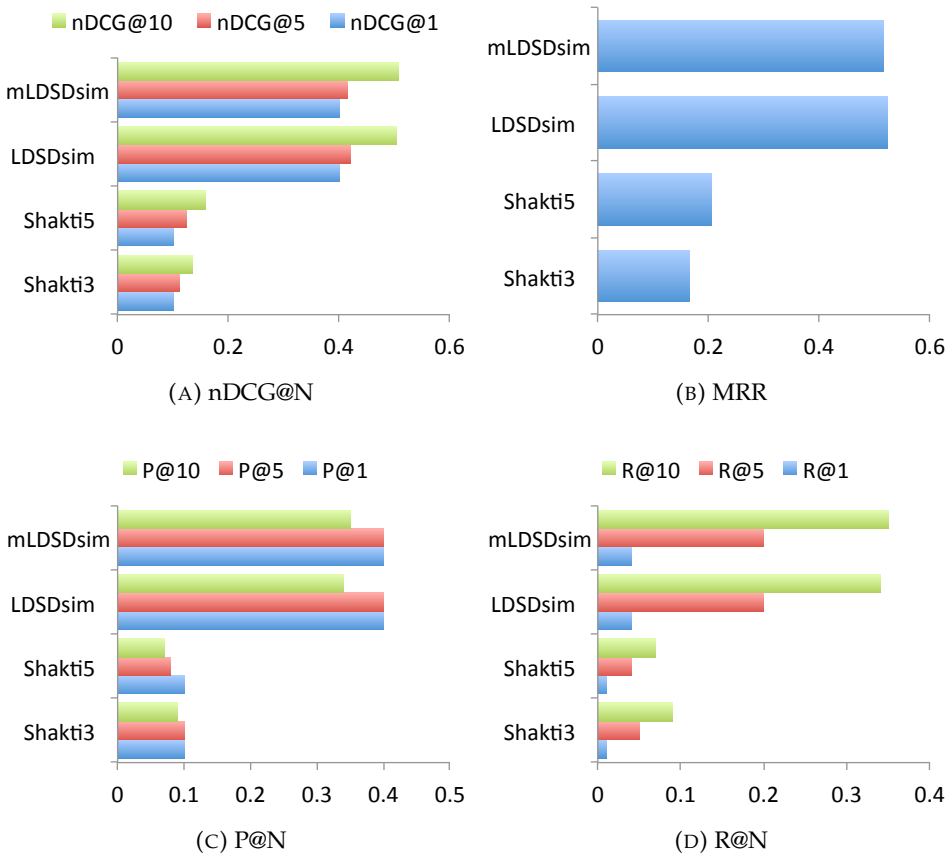


FIGURE 5.2: The results of recommendations for 10 random samples in the music domain in terms of different evaluation metrics.

5.4.1 Incorporating the Number of Linked Resources via A Link

In *LDSD*, C_i (or C_o) equals 1 if there is an entity r_n linked to r_a and r_b via an incoming (or outgoing) predicate l_x. In contrast, we modify the distance measure based on the intuition that two entities are more similar if there are a greater number of linked entities via a predicate l_x. For instance, if two music artists have 10 `dbo:MusicalArtist`(s) in common via the predicate `dbo:associatedMusicalArtist`, then they are more similar than two other music artists that have only one `dbo:MusicalArtist` in common via the same predicate.

The modified semantic distance measure based on this intuition is defined as follows, which is denoted by $mLDSD'_{\beta}$ (Equation 5.6).

$$mLDSD'_{\beta}(r_a, r_b) = \frac{1}{1 + \sum_x \frac{C_d(l_x, r_a, r_b)}{1+\log(C_d(l_x, r_a))} + \sum_x \frac{C_d(l_x, r_b, r_a)}{1+\log(C_d(l_x, r_b))} + \sum_x \frac{C'_i(l_x, r_a, r_b)}{1+\log(\frac{C_i(l_x, r_a)+C_i(l_x, r_b)}{2})} + \sum_i \frac{C'_o(l_x, r_a, r_b)}{1+\log(\frac{C_o(l_x, r_a)+C_o(l_x, r_b)}{2})}} \tag{5.6}$$

where C'_i (or C'_o) of $mLDSD'_{\beta}$, is equal to the number of entities linked to r_a and r_b via an incoming (or outgoing) predicate l_x.

The normalizations of $C'_i(\cdot)$ and $C'_o(\cdot)$ are carried out by considering both entities r_a and r_b in $mLDSD'_{\beta}$, which is the same as $mLDSD'$ and also satisfies the symmetry property. That is, the normalization of $C'_i(\cdot)$ is carried out by the average of $C_i(l_x, r_a)$ and $C_i(l_x, r_b)$. In contrast, the normalization of $C_i(\cdot)$ in $LDSD$ is carried out by considering the first entity r_a only. Similarly, the normalization of $C'_o(\cdot)$ is carried out using the average of $C_o(l_x, r_a)$ and $C_o(l_x, r_b)$ for $mLDSD'_{\beta}$.

5.4.2 Applying Global Normalizations

Both $LDSD$, $mLDSD'$ and $mLDSD'_{\beta}$ use local normalizations, i.e., normalizations are carried out in the local context of r_a and r_b. Instead of using local normalizations, here we use global normalizations of a path to investigate the impact on calculating the distance between two entities. The distance measure can be defined as Equation 5.7 and we use $mLDSD'_{\gamma}$ to refer to the distance measure in the rest of the thesis.

$$mLDSD'_{\gamma}(r_a, r_b) = \frac{1}{1 + \sum_x \frac{C_d(l_x, r_a, r_b)}{1+\log(C_{dp}(l_x))} + \sum_i \frac{C_d(l_x, r_b, r_a)}{1+\log(C_{dp}(l_x))} + \sum_x \sum_j \frac{C_i(l_x, r_j, r_a, r_b)}{1+\log(C_{ip}(l_x, r_j))} + \sum_x \sum_j \frac{C_o(l_x, r_j, r_a, r_b)}{1+\log(C_{op}(l_x, r_j))}} \tag{5.7}$$

In $LDSD$, the normalizations of $C_d(l_x, r_a, r_b)$ and $C_d(l_x, r_b, r_a)$ are carried out using $C_d(l_x, r_a)$ that computes the number of entities r_n from r_a via l_x (see Figure 5.3). In contrast, $mLDSD'_{\gamma}$ penalizes the importance of a path between two entities according to the global appearances of the path in the whole DBpedia graph. For example, in $mLDSD'_{\gamma}$, the normalizations of C_d functions are carried out using $C_{dp}(l_x)$ that computes the global appearances of the path $[l_x]$ between any two entities in DBpedia (see Figure 5.4).

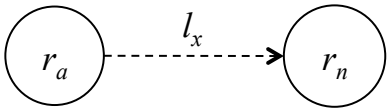


FIGURE 5.3: Local normalization of C_d function in Equations 5.1, 5.4 and 5.6: the number of entities from r_a to r_n via l_x.

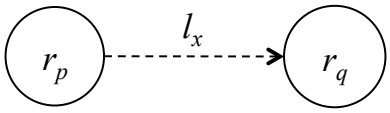


FIGURE 5.4: Global normalization of C_d function in Equation 5.7: the number of appearances from r_p to r_q via l_x in a graph.

Furthermore, for indirect paths between two resources, $mLDSD'_\gamma$ normalizes each indirect path by the number of global appearances of such an indirect path. Taking incoming indirect paths for example, $C_i(l_x, r_j, r_a, r_b)$ equals 1 if there is a path $\left[\overset{l_x}{\leftarrow}, r_j, \overset{l_x}{\rightarrow}\right]$ from r_a to r_b, and 0 if not. The normalization of $C_i(l_x, r_j, r_a, r_b)$ is then carried out using $C_{ip}(l_x, r_j)$ that computes the global appearances of the path $\left[\overset{l_x}{\leftarrow}, r_j, \overset{l_x}{\rightarrow}\right]$ between any two entities in DBpedia (see Figure 5.6).

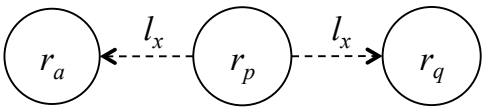


FIGURE 5.5: Local normalization of C_i function in Equations 5.1, 5.4 and 5.6: the number of entities linked to a resource via incoming predicate l_x as r_a.

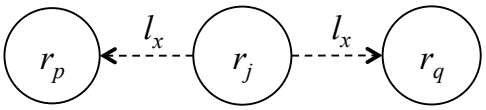


FIGURE 5.6: Global normalization of C_i function in Equation 5.7: the number of appearances of the path from r_p to r_q via the path $\left[\overset{l_x}{\leftarrow}, r_j, \overset{l_x}{\rightarrow}\right]$.

5.5 Evaluation

In this section, we describe a Facebook dataset used in addition to the Last.fm dataset for our experiment, and the methods compared for evaluating our proposed measures.

5.5.1 Facebook Dataset

The Facebook dataset[2] was collected from Facebook profiles about personal preferences ("likes") in the music domain, which consists of 52,072 users and 21 liked items on average. The items available in the dataset have been mapped to their corresponding DBpedia URIs. We randomly select 500 users with 10,590 preference records for the experiment. The main details of the dataset and its subset for our experiment are presented in Table 5.3.

TABLE 5.3: Descriptive statistics of the Facebook dataset.

Dataset	# of users	# of items	# of liked items			
			min.	max.	avg.	std.
Total	52,072	6,375	15	37	21	6.20
Subset	500	2,566	15	37	21	6.18

As we are interested in to what extent different distance measures perform on *cold-start* situations in recommender systems, we randomly selected one item that a user has liked for constructing our training set, and the rest of the liked items from users were used for constructing the test set. The candidate set consists of all items in the test set, which results in 2,566 distinct items in total.

Therefore, the task is providing the top-N recommendations from items in the candidate set based on an item that each user has liked. The preference score of each candidate item with respect to a user was measured by similarity/distance measures based on background knowledge about a candidate item and the one item that the user has liked.

5.5.2 Compared Methods

We compare the aforementioned semantic similarity/distance measures for calculating the similarity between two entities r_a and r_b in the context of recommender systems. We exclude *Shakti* in this experiment as we have shown it performed worst compared to *LDSD* and $mLDSD'$ in the preliminary evaluation (see Figure 5.2).

- *mLDSD* (Equation 5.1). This is the Linked Data Semantic Distance Measure proposed in Passant, 2010a.
- $mLDSD'$ (Equation 5.4). The modified *LDSD* introduced in Section 5.2.2 in order to satisfy the symmetry property.

[2] https://2015.eswc-conferences.org/important-dates/call-RecSys.html

- $mLDSD'_{\beta}$ (Equation 5.6). The modified *LDSD* introduced in Section 5.4 with a normalization strategy considering both r_a and r_b, and that also considers the number of entities linked to r_a and r_b via each predicate for measuring the similarity between them.
- $mLDSD'_{\gamma}$ (Equation 5.7). The modified *LDSD* introduced in Section 5.4 with a global normalization strategy, which considers the number of entities linked to r_a and r_b for measuring the similarity between them.

5.5.3 Results

Table 5.4 and Table 5.5 show the recommendation performance on the Last.fm and Facebook datasets, respectively, using the four compared methods in terms of the same evaluation metrics used in the preliminary evaluation.

TABLE 5.4: The results of item recommendations using different semantic similarity/distance measures on the Last.fm dataset. The best performance in terms of each evaluation metric is in **bold**.

	$LDSD$	$mLDSD'$	$mLDSD'_{\beta}$	$mLDSD'_{\gamma}$
MRR	0.5241	0.5157	0.6251	**0.6355**
nDCG@1	0.4000	0.4000	0.5000	**0.5000**
nDCG@5	0.4199	0.4146	**0.5319**	0.5294
nDCG@10	0.5042	0.5069	0.5884	**0.6257**
P@1	0.4000	0.4000	0.5000	**0.5000**
P@5	0.4000	0.4000	0.4400	**0.4400**
P@10	0.3400	0.3500	0.3500	**0.4100**
R@1	0.0400	0.0400	0.0500	**0.0500**
R@5	0.2000	0.2000	0.2200	**0.2200**
R@10	0.3400	0.3500	0.3500	**0.4100**

On the Last.fm dataset, overall, $mLDSD'_{\gamma}$ provides the best performance (Table 5.4). However, due to the small size of the manually constructed dataset, it is difficult to recognize any statistical significance between different measures. In this regard, we focus on the results of item recommendations on the Facebook dataset (Table 5.5) using different semantic similarity/distance measures.

As we can see from Table 5.5, $mLDSD'_{\gamma}$ provides the best performance followed by $mLDSD'_{\beta}$. Similar to the results of recommendations on the

TABLE 5.5: The results of item recommendations using different semantic similarity/distance measures on the Facebook dataset. The best performance in terms of each evaluation metric is in **bold**.

	$LDSD$	$mLDSD'$	$mLDSD'_\beta$	$mLDSD'_\gamma$
MRR	0.6447	0.6448	0.8718	**0.8875**
nDCG@1	0.4880	0.4840	0.8540	**0.8720**
nDCG@5	0.5672	0.5706	0.6893	**0.7052**
nDCG@10	0.6126	0.6181	0.7287	**0.7484**
P@1	0.4880	0.4840	0.8540	**0.8720**
P@5	0.2020	0.2044	0.2240	**0.2324**
P@10	0.1282	0.1306	0.1382	**0.1450**
R@1	0.0234	0.0232	0.0412	**0.0420**
R@5	0.0484	0.0490	0.0542	**0.0560**
R@10	0.0611	0.0620	0.0668	**0.0698**

Last.fm dataset, $LDSD$ and $mLDSD'$ have similar performance in terms of all evaluation metrics. Both $mLDSD'_\beta$ and $mLDSD'_\gamma$, which incorporate the number of linked resources via a link, outperform $LDSD$ and $mLDSD'$ significantly. For example, the recommendation performance was improved by 37.7%, 78.7%, and 79.5% in terms of MRR, nDCG@1 (P@1), and R@1, respectively. Another observation is that the semantic distance measure using a global normalization strategy ($mLDSD'_\gamma$) performs significantly better than the one using a local normalization strategy ($mLDSD'_\beta$).

The results indicate that incorporating the number of linked resources and adopting different normalization strategies such as local normalizations by considering both resources ($mLDSD'_\beta$), and the global normalizations of paths ($mLDSD'_\gamma$) can improve the performance of the recommender system. In addition, the best performance achieved by $mLDSD'_\gamma$ indicates that the global normalizations of paths between entities represent the importance of paths better than local ones.

5.6 Study of Linked Data Sparsity Problem

During the experiment mentioned in the previous section using the Last.fm dataset, we found that some of the random samples with less incoming/outgoing links yielded poor recommendation performance. For instance, the nDCG@10 of recommendations for `dbr:Jasmin_Thompson` is 0.46, which is one of the random samples that has 42 outgoing links and 3 incoming links. In contrast, the nDCG@10 of recommendations for `dbr:Dead_Kennedys` is 0.9, which has 117 outgoing links and 119 incoming links.

This observation motivates us to investigate whether the performance of the LOD-enabled recommender system based on semantic similarity/distance measures suffers from "Linked Data sparsity". Here, the *Linked Data sparsity problem* means that the performance of the recommender system based on semantic similarity measures decreases when entities lack information (i.e., when they have a lesser number of incoming/outgoing relationships to other entities). In this regard, the null hypothesis to test can be defined as below:

H_0 : *The number (log scale) of incoming/outgoing links for entities has no relationship to the performance of a recommender system based on semantic similarity/distance measures.*

We use the logarithm of the number, which is denoted as number (log scale), to decrease the variation in numbers. We reject the null hypothesis if the number (log scale) of incoming/outgoing links and the nDCG of recommendations have a strong relationship (*Pearson's correlation > 0.5*), otherwise we accept the null hypothesis.

To this end, we additionally selected 10 popular DBpedia entities of type `dbo:MusicArtist` as samples, and then calculated the nDCG at 1, 5 and 10 in the same way as we did for the 10 randomly selected samples. The assumption here is that the popular samples tend to have more information (i.e., incoming/outgoing links) than random samples. This is because these entities in DBpedia are a reflection of the corresponding concepts/articles in Wikipedia, and usually popular music artists have more information thanks to a higher number of contributors.

First, we intend to see whether the recommendation system performs better on popular samples than on random ones. On top of that, we aim to investigate the correlation by calculating the *Pearson's coefficient* between the number (log scale) of incoming/outgoing links for entities and the nDCG of the recommender system.

Figure 5.7 shows the nDCG@N results for random and popular samples. As we can see from the figure, the nDCG results of the recommender system on popular samples are significantly better than the results on random ones. Following this finding, we calculate the correlation between the number (log scale) of incoming/outgoing links for entities and the performance (nDCG) of the recommender system. We report nDCG@10 based on $mLDSD'_{\gamma}$ here, and similar results can be observed by using other measures.

The result shows the performance of the recommender system has a very strong positive relationship (Figure 5.8, *Pearson's correlation of 0.579*) with the total number (log scale) of incoming/outgoing links ($p < 0.01$). Hence, the null hypothesis is rejected. In other words, the performance of the recommender system decreases for the resources with sparsity (i.e., less incoming/outgoing links). It also indicates that, on the one hand, utilizing Linked Data to build a recommender system can mitigate the traditional sparsity problem (Heitmann and Hayes, 2010) of collaborative recommender systems, but on the other hand, the system can also have a *Linked Data sparsity problem* for entities in the Linked Data set that the recommender system has adopted.

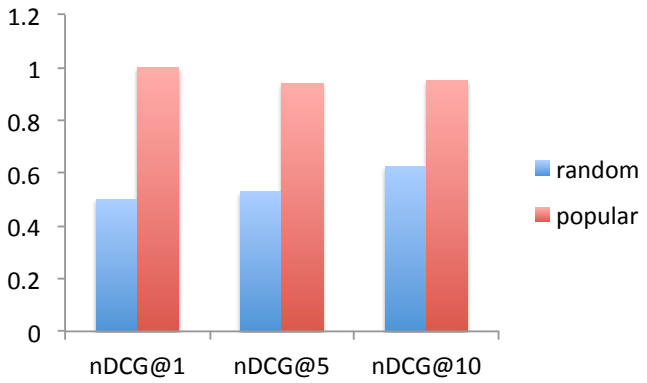


FIGURE 5.7: The recommendation performance in terms of nDCG@N on random samples and popular ones.

5.7 Summary

In this chapter, we proposed a semantic similarity/distance measure named $mLDSD_{sim}$ for calculating the similarity between two entities in a knowledge graph such as DBpedia. In addition, we proposed different linked data semantic distance measures for $mLDSD_{sim}$ (Section 5.4), and evaluated those measures compared to *Shakti* and *LDSD* (Section 5.5). Results show that using our proposed approach can significantly improve the recommendation performance in terms of all evaluation metrics compared

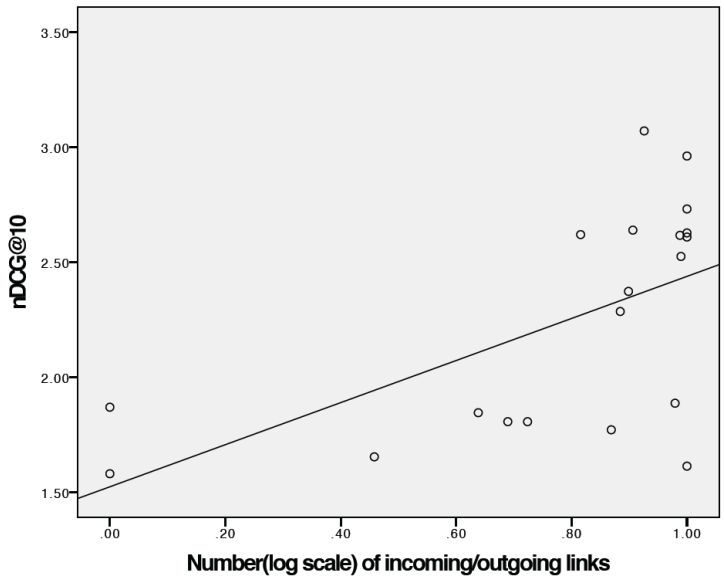


FIGURE 5.8: Scatter plot of nDCG@10 and the number (log scale) of links, r=0.579.

to using other semantic similarity/distance measures. We found that (1) incorporating the number of entities via indirect paths, and (2) using a global normalization strategy based on the global appearances of paths improve the performance of $mLDSD_{sim}$ significantly in the context of recommender systems. The implementation of $mLDSD_{sim}$ is available from https://github.com/parklize/resim/, which can be used with either a public SPARQL endpoint or the HDT (Fernández et al., 2013) dump of a knowledge graph such as DBpedia or Wikidata (Vrandečić and Krötzsch, 2014).

In Section 5.6, we investigated whether the performance of a LOD-enabled recommender system, which adopts similarity measures for calculating the similarity between items (entities), suffers from the "Linked Data sparsity problem". The results show that the performance of the recommender system has a very strong positive relationship with the number (log scale) of the total number of incoming/outgoing links ($p < 0.01$) for entities.

Chapter 6

Semantics-Aware Machine Learning Approaches for Item Recommendations

In previous chapters, we have focused on the *cold-start* problem in recommender systems on OSNs using *top-down* approaches where collaborative filtering approaches are not working due to the lack of training data. When there is a large amount of explicit feedback (training data) available, collaborative filtering approaches such as *matrix factorization* have been widely used for learning the latent representations of users and items in order to provide personalized recommendations.

This chapter mainly focuses on *collaborative filtering* and *bottom-up* semantics-aware approaches for learning the semantic representations of users and items in latent dimensions for item recommendations based on explicit feedback from users and the background knowledge about those items. For example, given 10 musical artists liked by users on average, how (can) we build a recommendation model based on this explicit feedback with respect to musical artists and the background knowledge about those artists in DBpedia as training dataset (see Figure 6.1). The main contributions of this chapter have been published in Piao and Breslin, 2017a and Piao and Breslin, 2018f.

6.1 Introduction

As discussed in Section 2.4, there has been different types of approaches for consuming background knowledge about items together with explicit feedback for collaborative filtering. Some previous studies compared their

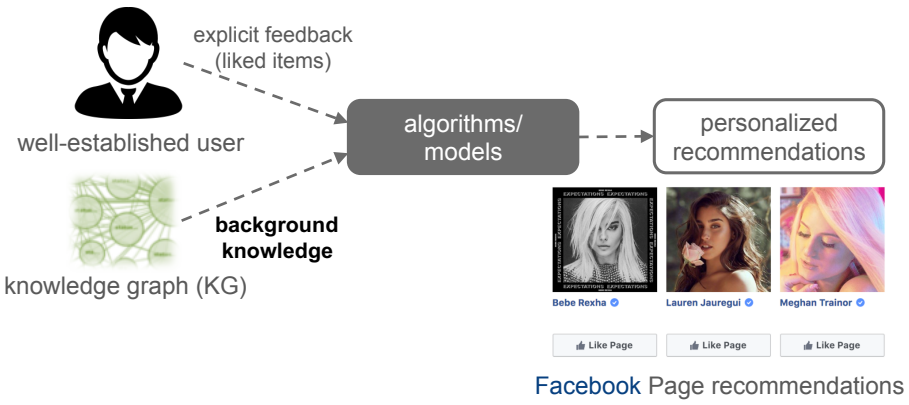


FIGURE 6.1: Overview of semantics-aware machine learning approaches for LODRecSys, which are based on explicit feedback (e.g., liked items) and the background knowledge about those items for learning a recommendation model.

LODRecSys approaches against well-established *collaborative filtering* approaches such as *kNN* and *matrix factorization* models such as BPRMF (Rendle et al., 2009), and have shown the benefits of consuming background knowledge powered by LOD. On the other hand, *matrix factorization* models such as *BPRMF*, which do not exploit LOD-enabled features, have shown competitive performance even compared to some LODRecSys approaches (Musto et al., 2016b; Noia et al., 2016). This has in turn motivated us to investigate factorization models consuming LOD-enabled features.

In this chapter, we propose *LODFM* which leverages lightweight LOD features with FMs for Linked Open Data-enabled recommender systems. In addition, we study transfer learning between item recommendations and the KG completion task in order to incorporate the incompleteness of a KG. The contributions of this chapter are summarized as follows.

- We investigate lightweight LOD-enabled features (Section 6.2.2), which can be directly obtained via the public DBpedia SPARQL endpoint, for a factorization machine to provide the top-N recommendations. Therefore, there is no need to construct a graph which combines user-item interactions (e.g., *likes*, *dislikes*) and background knowledge about items. In addition, we investigate to what extent different sets of these features contribute to factorization machines in terms of recommendation performance.

- In Section 6.2.5, we comprehensively evaluate our approach by comparing it to other approaches such as *PopRank*, *kNN*, *BPRMF*, and a state-of-the-art LODRecSys approach *SPRank* (Noia et al., 2016) in

terms of five different evaluation metrics.

- We study knowledge transfer between two tasks: (1) *item recommendations*, and (2) *KG completion* for the specific domain of items, via a co-factorization model (*CoFM*). This transfer learning model incorporates the incompleteness of a KG for item recommendations, and incorporates the knowledge from item recommendations for completing the KG (Section 6.3.1).
- In Section 6.4, we evaluate *CoFM* with three baselines for each task, and show that incorporating the *incompleteness* of a KG outperforms the baselines significantly. In addition, we show that exploiting the knowledge from item recommendations improves the performance of KG completion with respect to the domain of items, which has not been studied in previous studies.

6.2 Factorization Machines Leveraging Lightweight LOD-enabled Features

In this section, we first briefly introduce factorization machines (FMs) (Rendle, 2010) and the optimization criteria we used in this study (Section 6.2.1). Next, we will describe our features from user-item interactions as well as background knowledge from DBpedia (Section 6.2.2). Sections 6.2.3 and 6.2.4 describe datasets for our experiments and baselines for comparison, respectively. Finally, we discuss the experimental results in Section 6.2.5.

6.2.1 Proposed Approach

Factorization machines, which can mimic other well known factorization models such as *matrix factorization*, *SVD++* (Koren, 2009), have been widely used for collaborative filtering tasks (Rendle, 2012). FMs are able to incorporate the high-prediction accuracy of factorization models and flexible feature engineering. An important advantage of FMs is the model equation:

$$\hat{y}^{FM}(x) = w_0 + \sum_{i=1}^{p} w_i x_i + \sum_{i=1}^{p} \sum_{j=i+1}^{p} < \boldsymbol{v_i}, \boldsymbol{v_j} > x_i x_j \tag{6.1}$$

where $w_0 \in \mathbb{R}$ denotes a bias term, $\boldsymbol{x} \in \mathbb{R}^p$ and $\boldsymbol{w} \in \mathbb{R}^p$ denote input variables and their corresponding weights, and $\boldsymbol{v_i} \in \mathbb{R}^m$ denotes the latent factors of i-th variable. The first part of the FM model captures the interactions of each input variable x_i, while the second part of it models all pairwise

interactions of input variables $x_i x_j$. Each variable x_i has a latent factor v_i, which is an m-dimensional vector that allows FMs to work well even in highly sparse data. In order to learn the parameters such as latent factors with respect to users and items in FMs, we have to define an optimization function with respect to the training dataset.

Optimization. In this work, we use a widely used *pairwise* optimization approach - Bayesian Personalized Ranking (BPR). The loss function was proposed by Rendle et al., 2009.

$$l(x_1, x_2) = \sum_{x_1 \in C_u^+} \sum_{x_2 \in C_u^-} (-\log[\delta(\hat{y}^{FM}(x_1) - \hat{y}^{FM}(x_2))]) \tag{6.2}$$

where δ is a sigmoid function: $\delta(x) = \frac{1}{1+e^{-x}}$, and C_u^+ and C_u^- denote the set of positive and negative feedback items respectively. $\hat{y}^{FM}$ is the predicted score for a given item. L2-regularization is used for the loss function. In detail, a positive training instance consists of a user and an item which the user liked in the training dataset. A negative instance for the user consists of the user and a randomly chosen item which is not in the list of items the user liked before in the training set. The intuition behind BPR is that a liked item for a user should be ranked higher (with a higher score) compared to a random one in the list of items with which the user has not interacted.

Learning. We use the well-known *stochastic gradient descent* (SGD) algorithm to learn the parameters in our model. To avoid overfitting on the training dataset, we adopt an early stopping strategy as follows.

1. Split the dataset into training and validation sets.
2. Measure the current loss on the validation set at the end of each epoch.
3. Stop and remember the epoch if the loss has increased.
4. Re-train the model using the whole dataset.

6.2.2 LOD-enabled Features

Figure 6.2 presents an overview of features for our FM. The details of each set of features are described below.

User and item index. The first two sets of features indicate the indexes of the user and item in a training example. A feature value equals 1 for the corresponding user/item index. For example, $val(U_i) = 1$ and $val(I_j) = 1$ denote an example about the i-th user and j-th item.

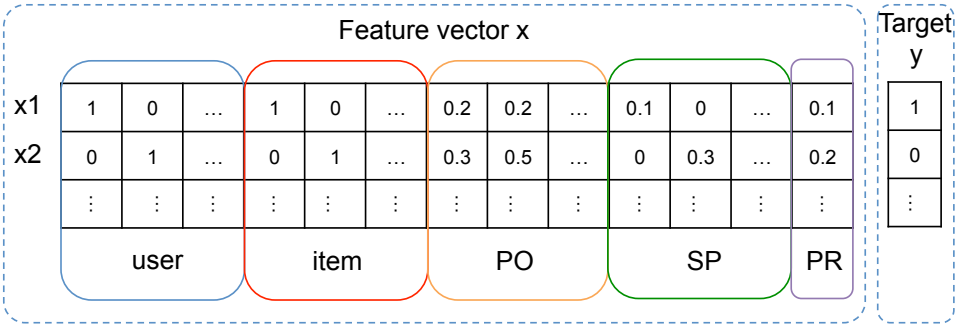


FIGURE 6.2: Overview of features for a factorization machine. PO denotes all predicate-objects, and SP denotes all subject-predicates for items in the dataset. PR denotes the PageRank scores of items.

Predicate-Object list (PO). This set of features denotes all predicate-objects of an item i when i is a subject in RDF triples. This set of features can be obtained easily via the DBpedia SPARQL endpoint by using a SPARQL query as shown below.

```
PREFIX dbo:<http://dbpedia.org/ontology/>
PREFIX dct:<http://purl.org/dc/terms/>

SELECT DISTINCT ?p ?o WHERE { { <itemURI> ?p ?o  .
FILTER REGEX(STR(?p), ''^http://dbpedia.org/ontology'') .
FILTER (STR(?p) NOT IN (dbo:wikiPageRedirects,
                        dbo:wikiPageExternalLink)) .
FILTER ISURI(?o) }
UNION { <itemURI> ?p ?o .
FILTER ( STR(?p) IN (dct:subject) ) } }
```

LISTING 6.1: SPARQL for extracting PO features.

An intuitive way of giving feature values for a PO might be to assign 1 for all predicate-objects of an item i (PO_i). However, it can be biased as some entities in DBpedia have a great number of predicate-objects while others do not. Therefore, we normalize the feature values of PO_i based on the size of PO_i so that all the feature values of PO_i sum up to 1. Formally, the feature value of the j-th predicate-object for an item i is measured as $val(PO_i(j)) = \frac{1}{|PO_i|}$. Take the graph in Figure 6.3 as an example, as we have two predicate-objects for the movie `dbr:The_Godfather`, where each predicate-object of the movie will have a feature value of 0.5 (see Figure 6.4).

Subject-Predicate list (SP). Similar to the PO, we can obtain incoming background knowledge about an item i where i is an object in RDF triples. This

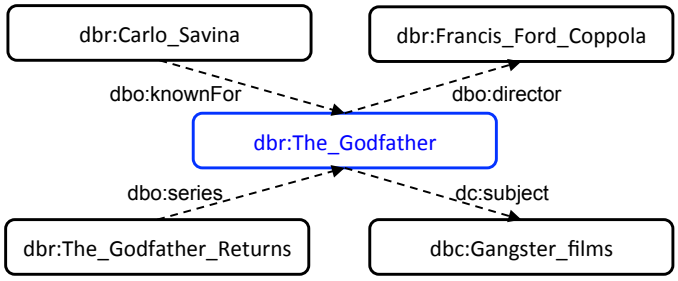


FIGURE 6.3: An example of background knowledge about the movie "*The Godfather*" from DBpedia.

...	0.5	...	0.5	...
...	dbo:director➔dbr:Francis_Ford_Coppola	...	dc:subject➔dbc:Gangster_films	...

FIGURE 6.4: An example of PO values for the movie entity `dbr:The_Godfather` in Figure 6.3.

set of features can be obtained by using a SPARQL query as shown in Listing 6.2.

```
PREFIX dbo:<http://dbpedia.org/ontology/>

SELECT DISTINCT ?s ?p WHERE { ?s ?p <itemURI> .
FILTER REGEX(STR(?p), ''^http://dbpedia.org/ontology'') .
FILTER (STR(?p) NOT IN (dbo:wikiPageRedirects,
                        dbo:wikiPageExternalLink,
                        dbo:wikiPageDisambiguates) }
```

LISTING 6.2: SPARQL for extracting SP features.

In the same way as we normalized feature values of PO_i for an item i, we normalize the feature values of SP_i based on the size of SP_i so that all the feature values of SP_i sum up to 1. The feature value of the j-th SP for an item i is measured as $val(SP_i(j)) = \frac{1}{|SP_i|}$.

PageRank score (PR). PageRank (Page et al., 1999) is a popular algorithm with the purpose of measuring the relative importance of a node in a graph. In order to capture the importance of an entity in Wikipedia/DBpedia, Thalhammer and Rettinger, 2016 proposed providing PageRank scores of all DBpedia entities, which are based on links using `dbo:wikiPageWikiLink`(s) among entities. A PageRank score of an item (entity) can be a good indicator of the importance of an entity for recommendations in our case. The PageRank score of a DBpedia entity can be obtained by using the SPARQL query

as shown in Listing 6.3.

```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbo:<http://dbpedia.org/ontology/>
PREFIX vrank:<http://purl.org/voc/vrank#>

SELECT ?score FROM <http://dbpedia.org>
FROM <http://people.aifb.kit.edu/ath/#DBpedia_PageRank>
WHERE
{ <itemURI> vrank:hasRank/vrank:rankValue ?score . }
```

LISTING 6.3: SPARQL for extracting PR features.

The scale of PageRank scores is different from other feature values, which can delay the convergence of learning parameters for our model. In this regard, we normalize the PageRank scores by their maximum value.

$$val(PR_i) = \frac{PageRank_i}{max(PageRank_j, j \in I)} \tag{6.3}$$

where $PageRank_i$ denotes the original PageRank score of i which is obtained from the SPARQL endpoint, and $max(PageRank_j, j \in I)$ denotes the maximum PageRank score of all items.

6.2.3 Datasets

We used two datasets for our experiments; one is the Facebook dataset introduced in Section 5.5.1, which consists of 52,072 users and 21 liked items on average in the music domain, and the other is the same mapped Movielens dataset used in Noia et al., 2016. The mapped Movielens dataset was originally derived from the Movielens dataset[1], which consists of users and their ratings about movie items. To facilitate LODRecSys, each of the items in this dataset has been mapped into DBpedia entities if there is a mapping available[2].

In the same way as Noia et al., 2016, we consider ratings higher than 3 as positive feedback and others as negative ones. Table 6.1 shows details about the dataset. The dataset consists of 3,997 users and 3,082 items with 827,042 ratings where 56% of them are positive ratings. For both datasets, we split each one into training (80%) and test (20%) sets for our experiment.

[1] https://grouplens.org/datasets/movielens/1m/
[2] http://sisinflab.poliba.it/semanticweb/lod/recsys/datasets/

TABLE 6.1: Movielens dataset statistics

# of users	3,997
# of items	3,082
# of ratings	827,042
avg. # of ratings	206
sparsity	93.27%
% of positive ratings	56%

6.2.4 Compared Methods

We use four approaches including a baseline *PopRank* and other methods which have been frequently used in the literature (Musto et al., 2016b; Noia et al., 2016) to evaluate our proposed method.

- *PopRank:* This is a non-personalized baseline approach which recommends items based on the popularity of each item.
- *kNN-item: (kNN)* This is a collaborative filtering approach based on the k most similar items. We use a MyMedialite (Gantner et al., 2011) implementation for this baseline where $k = 80$.
- *BPRMF* (Rendle et al., 2009): This is a matrix factorization approach for learning latent factors for users and items. We use a MyMedialite implementation for this baseline where the dimensionality of the factorization $m = 200$.
- *SPRank* (Noia et al., 2016): This is a *learning-to-rank* approach for LOD-RSs based on *semantic paths* extracted from a graph including user-item interactions (e.g., likes, dislikes, etc.) as well as the background knowledge obtained from DBpedia. In detail, *semantic paths* are sequences of predicates including *likes* and *dislikes* based on user-item interactions. For example, given the graph information `user1` $\rightarrow$ `likes` $\rightarrow$ `item1` $\rightarrow$ `p1` $\rightarrow$ `item2`, a semantic path (`likes`, `p1`) can be extracted from `user1` to `item2`.

  The difference between *SPRank* (Noia et al., 2016) and our approach in terms of features is that the authors considered predicate-objects for each item, including the predicate `dbo:wikiPageWikiLink` which cannot be queried via the DBpedia Endpoint, but requires setting up a local endpoint using a DBpedia dump. In contrast, we only consider sets of LOD-enabled features which can be obtained from a public DBpedia Endpoint. We use *LMART* (Wu et al., 2010) as the learning algorithm for *SPRank* as this approach overall provides the best performance compared to other learning-to-rank algorithms in

(Noia et al., 2016). We used the author's implementation[3] which has been optimized for nDCG@10.

SPRank requires ratings or binary response (e.g., likes or dislikes) for items in order to construct a ranked list to run the learning-to-rank algorithms. As the Facebook dataset only contains a set of liked items for each user, we randomly selected x items that a user u has not interacted with, where x is the size of liked items of u. The intuition is similar to BPR, i.e., an item liked by a user should be ranked higher than a random item which is not in the set of items that the user liked.

6.2.5 Results

In this section, we first compare our approach to the aforementioned methods in terms of four evaluation metrics introduced in Section 5.3.1: MRR, nDCG@N, P@N, and R@N. We denote our approach as *LODFM*, and the results of *LODFM* are based on best tuned parameters, i.e., $m = 200$ using PO and PR as LOD-enabled features. We then discuss self comparison by using different sets of features, as well as a different dimensionality m for factorization.

Comparison with baselines. The results of comparing our proposed approach with the baselines on the Facebook and Movielens datasets are presented in Table 6.2 and 6.3, respectively. We observe similar trends on both datasets. The baseline method *PopRank* does not perform well compared to other approaches, and *BPRMF* provides the best performance among the methods compared. Overall, *LODFM* provides the best performance in terms of all evaluation metrics. In line with the results from Noia et al., 2016, *SPRank* does not perform as well on the Movielens dataset compared to other collaborative filtering approaches such as *kNN* and *BPRMF*. Similar results can be observed on the Facebook dataset.

On the Facebook dataset, we observe that *LODFM* significantly outperforms *SPRank* as well as other baseline methods. For example, a significant improvement of *LODFM* over *BPRMF* in MRR (+1.8%) can be observed. On the Movielens dataset, *kNN* is the best performing method among the baseline methods in terms of P@5 and P@10 while *BPRMF* is the best performing baseline in terms of other evaluation metrics on the Movielens dataset. A significant improvement of *LODFM* over *BPRMF* in MRR (+5.3%), nDCG@10 (+4.6%), P@10 (+12.9%) and R@10 (+8%) can be noticed. These results indicate that LOD-enabled features are able to improve the recommendation performance for factorization models.

[3] https://github.com/sisinflab/lodreclib

TABLE 6.2: Recommendation performance compared to baselines in terms of five different evaluation metrics on the Facebook dataset. The best performing strategy is in bold.

	PopRank	*kNN*	*BPRMF*	*SPRank*	*LODFM*
MRR	0.1253	0.2025	0.2132	0.0839	**0.2171**
nDCG@1	0.0488	0.0883	0.0936	0.0294	**0.0942**
nDCG@5	0.0973	0.1609	0.1681	0.0662	**0.1689**
nDCG@10	0.1332	0.2184	0.2285	0.0912	**0.2312**
P@1	0.0488	0.0883	0.0936	0.0294	**0.0942**
P@5	0.0401	0.0713	0.0768	0.0248	**0.0777**
P@10	0.0343	0.0605	0.0653	0.0214	**0.0667**
R@1	0.0121	0.0222	0.0235	0.0073	**0.0237**
R@5	0.0492	0.0891	0.0954	0.0306	**0.0969**
R@10	0.0841	0.1501	0.1612	0.0525	**0.1650**

TABLE 6.3: Recommendation performance compared to baselines in terms of five different evaluation metrics on the Movielens dataset. The best performing strategy is in bold.

	PopRank	*kNN*	*BPRMF*	*SPRank*	*LODFM*
MRR	0.4080	0.5756	0.5906	0.3013	**0.6218**
nDCG@1	0.2459	0.4086	0.4269	0.1758	**0.4685**
nDCG@5	0.2809	0.4049	0.4176	0.2195	**0.4537**
nDCG@10	0.3664	0.4753	0.5000	0.2845	**0.5231**
P@1	0.2459	0.4086	0.4269	0.1758	**0.4685**
P@5	0.2240	0.3538	0.3393	0.1287	**0.3829**
P@10	0.2104	0.3179	0.2883	0.1068	**0.3256**
R@1	0.0064	0.0132	0.0258	0.0082	**0.0268**
R@5	0.0305	0.0553	0.0977	0.0291	**0.1052**
R@10	0.0580	0.0978	0.1602	0.0488	**0.1730**

Compared to *kNN*, *LODFM* improves the performance by 8.2% and 2.4% in terms of P@5 and P@10, respectively. It is also interesting to observe that factorization models such as *BPRMF* and *LODFM* have much better

performance especially in terms of recall compared to *kNN* on the Movielens dataset. For example, *LODFM* improves the performance by 103%, 90% and 76.9% in terms of recall when N= 1, 5 and 10, respectively.

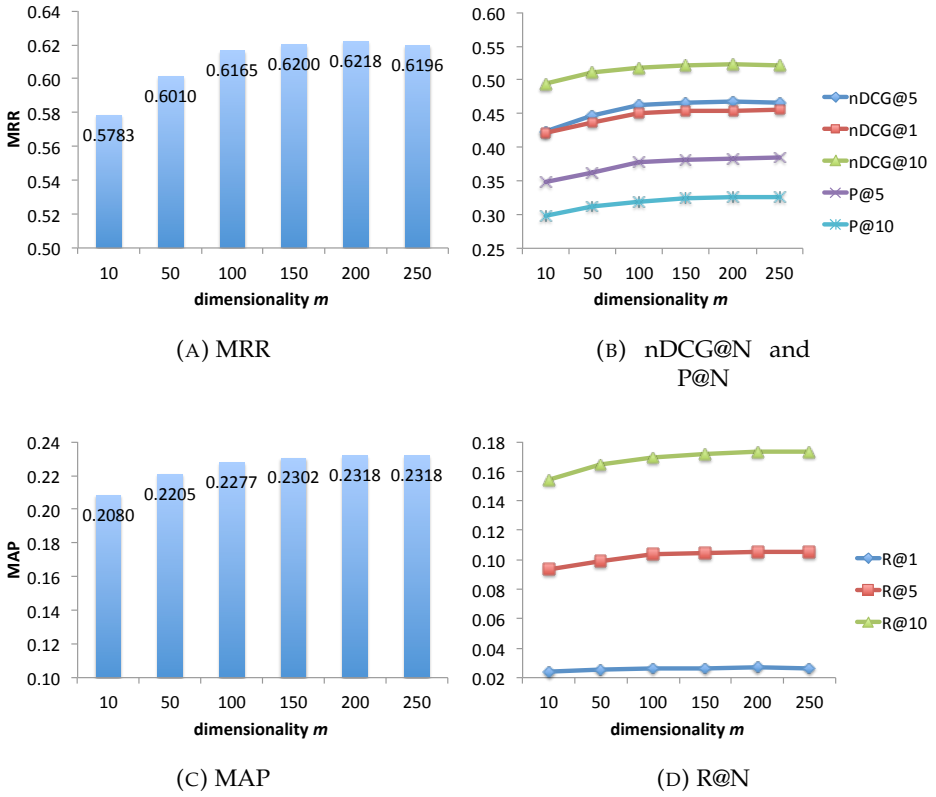
Analysis of features. To better understand the contributions of each feature set for recommendations, we discuss the recommendation performance on the Movielens dataset with different sets of features for the FM. Table 6.4 shows the recommendation performance of *LODFM* using different features with $m = 10$. As the focus here is to compare the performance in terms of different features, we used a fixed value 10 for the dimensionality m for reducing the experiment time. The two fundamental features - user and item indexes are included by default and omitted from the table for clarity.

TABLE 6.4: Recommendation performance of *LODFM* on the Movielens dataset using different sets of features such as predicate-object list (PO), subject-predicate list (SP) and PageRank scores (PR). The best performing strategy is in bold.

	PO	PO+SP	PO+PR	PO+SP+PR
MRR	0.5769	0.5403	**0.5783**	0.5561
nDCG@1	0.4224	0.3788	**0.4236**	0.3971
nDCG@5	0.4152	0.3861	**0.4214**	0.3963
nDCG@10	0.4904	0.4627	**0.4945**	0.4743
P@1	0.4224	0.3788	**0.4236**	0.3971
P@5	0.3459	0.3222	**0.3479**	0.3280
P@10	0.2973	0.2805	**0.2975**	0.2860
R@1	0.0237	0.0210	**0.0241**	0.0223
R@5	0.0931	0.0841	**0.0934**	0.0866
R@10	**0.1558**	0.1436	0.1541	0.1476

Overall, using a predicate-object list (PO) and the PageRank score (PR) of items provides the best performance compared to other strategies. As we can see from Table 6.4, PO+PR improves the recommendation performance compared to PO in terms of most of the evaluation metrics. Similar results can be observed by comparing PO+SP+PR against PO+SP, which shows the importance of PageRank scores of items. On the other hand, the performance is decreased by including SP, e.g., PO+SP vs. PO and PO+SP+PR vs. PO+PR. This shows that incoming knowledge about movie items is not helpful in improving recommendation performance in the context of using FMs.

Analysis of dimensionality m for factorization. The dimensionality of factorization plays an important role in capturing pairwise interactions of input variables when m is chosen to be large enough (Rendle, 2012). Figure 6.5 illustrates the recommendation performance using different values for the dimensionality of factorization using PO and PR as LOD-enabled features. The results of P@1 are equal to nDCG@1 and therefore omitted from Figure 6.5. As we can see from the figure, the performance consistently increases with higher values of m until $m = 200$ in terms of the five evaluation metrics. For example, the performance is improved by 7.5% in terms of MRR with $m = 200$ compared to $m = 10$. There is no significant improvement with values higher than 200 for m.



(A) MRR

(B) nDCG@N and P@N

(C) MAP

(D) R@N

FIGURE 6.5: Recommendation performance on the Movielens dataset based on different values for the dimensionality m of a FM using PO+PR in terms of different evaluation metrics.

6.3 Transfer Learning for Item Recommendations and Knowledge Graph Completion

In the previous section, we proposed *LODFM*, which leverages lightweight LOD-enabled features with FMs for LODRecSys. The focus of our approach

in Section 6.2, as well as previous studies leveraging KGs for recommender systems, has been on exploiting LOD-enabled features for different types of machine learning or graph algorithms.

Although previous studies have given some useful insights into leveraging background knowledge about items from KGs for recommender systems, most of these studies have not considered the *incompleteness* of KGs. A dedicated line of research has focused on the task of KG completion (Franz et al., 2009; Drumond et al., 2012), which can be categorized into two groups of embedding-based approaches. One is using factorization approaches such as tensor factorization (Bordes et al., 2013; Drumond et al., 2012; Nickel et al., 2016; Nickel et al., 2012), and the other is using neural network models (Guo et al., 2015; Wang et al., 2014; Ji et al., 2016).

Indeed, most KGs use the *Open World Assumption*, i.e., it is not necessarily false if a KG does not contain a certain piece of information. The piece of information may be true but is missing from the KG. For example, the piece of information with dotted lines in Figure 6.6 shows that the category `dbc`[4]`:Horror_films` is missing for the entity `dbr:Bled_White_(2011_film)` in DBpedia, which is important information in the context of recommending movies.

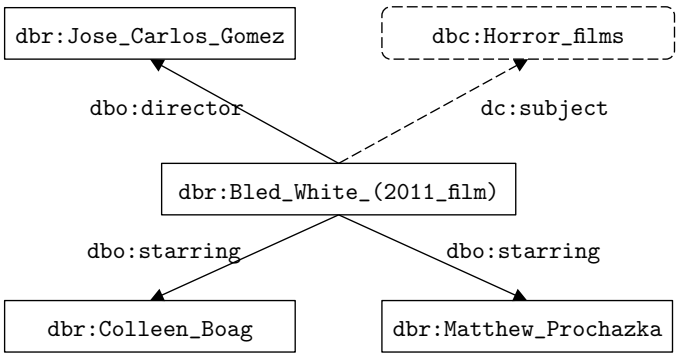


FIGURE 6.6: Pieces of information about the movie `dbr:Bled_White_(2011_film)` from DBpedia. The piece of information with dotted lines denotes missing information from the knowledge graph.

In this section, we leverage a co-factorization model to investigate transfer learning (Pan and Yang, 2010) between these two tasks: (1) *item recommendations*, and (2) *KG completion* with respect to the domain of items. Here, *transfer learning* denotes using one task as a "source" task and the other as a "target" task.

[4]The prefix `dbc` denotes http://dbpedia.org/resource/Category

First, with item recommendations as the target task and KG completion as the source task, we are interested in whether incorporating the *incompleteness* of a KG performs better when compared to *LODFM* which exploits existing knowledge from the KG, and outperforms other baselines. Secondly, we aim to investigate whether knowledge from item recommendations can be transferred to KG completion and improves the performance when KG completion is the target task.

6.3.1 Learning with a Co-Factorization Model

We begin by formulating the two tasks - (1) item recommendations, and (2) KG completion, and then describe state-of-the-art approaches for each task. Finally, we present a co-factorization model (*CoFM*) for transfer learning between these two tasks.

- *Item recommendations*: Given user-item interaction histories, i.e., *likes* or *dislikes* about items (we consider binary interactions in this study), our goal is to provide the top-N item recommendations for a target user.
- *KG completion*: This task can be formulated into a top-N recommendations task as well, in the same way as previous studies (Drumond et al., 2012; Ji et al., 2016). Here we are interested in a domain-specific KG, which consists of item-related triples where items are subjects in this domain-specific KG. For a given (*subject*, *predicate*) pair, the task is providing the top-N *object* recommendations from a set of candidate *objects*. Candidate *objects* are all objects in the *range* of a given *predicate* defined in the DBpedia ontology. For instance, given the *predicate* `dbo:starring`, the candidate objects consist of all entities with the type `dbo:Actor`, which is the *range* of the *predicate*[5].

Factorization machines for item recommendations

The first task is to provide the top-N item recommendations based on the history of user-item interactions. We use FMs (Equation 6.1) for item recommendations, and focus on a binary response $y_{d_{ui}}$ (e.g., a user u likes or dislikes an item i) for each item in this study. Let β_0 denote the bias, $\boldsymbol{\beta}_i$ denote the weights of features with respect to i, and $\boldsymbol{\theta}_i$ denote a list of latent factors for i, which can be learned through FMs with the training dataset. In addition, $\mathbf{x}_{\mathbf{d_{ui}}}$ denotes a list of explicit features in a training example d_{ui}. The simplest case for $\mathbf{x}_{\mathbf{d_{ui}}}$ is that it consists of one categorical feature to denote a user u, and the other categorical feature to denote an item i. Following

[5] http://dbpedia.org/ontology/starring

the definition of the FM model in the previous section (Algorithm 6.1), we can estimate the preference score of an item i based on $\mathbf{x_{d_{ui}}}$, $\boldsymbol{\beta}$ and $\boldsymbol{\Theta}$ as $f(s_{d_{ui}}|\mathbf{x_{d_{ui}}}, \boldsymbol{\beta}, \boldsymbol{\Theta})$, where

$$s_{d_{ui}} = \beta_0 + \boldsymbol{\beta}_u + \boldsymbol{\beta}_i + < \boldsymbol{\theta}_u, \boldsymbol{\theta}_i > \tag{6.4}$$

As discussed in the previous section, the task of recommending the top-N items can be formalized as optimizing the BPR (Rendle et al., 2009) loss as follows:

$$\ell(a_1, a_2) = \sum_{a_1 \in \mathcal{D}^+_{ui}} \sum_{a_2 \in \mathcal{D}^-_{ui}} -\log[\delta(s_{a_1} - s_{a_2})] \tag{6.5}$$

where δ is a sigmoid function: $\delta(x) = \frac{1}{1+e^{-x}}$, and $\mathcal{D}^+_{ui}$ and $\mathcal{D}^-_{ui}$ denote the set of positive and negative training instances, respectively. In fact, the FM using BPR for optimization with users and items as features is exactly a biased *BPRMF* (Rendle et al., 2009), which has been shown in the previous study (Rendle, 2012).

Translating embeddings for KG completion

The second task is the KG completion with respect to the domain of items, which can be formulated as *object* recommendations given a *subject* and *predicate* pair (Drumond et al., 2012). We use a translation-based embedding model, *TransE* (Bordes et al., 2013), for the second task. *TransE* is one of the most popular approaches for KG completion due to its effectiveness despite its simplicity.

The intuition behind this model is to learn latent factors for *subjects*, *predicates*, and *objects*, in order to satisfy $\boldsymbol{\phi}_s + \boldsymbol{\phi}_p \approx \boldsymbol{\phi}_o$ when (s, p, o) is a valid triple in the KG. In other words, for a valid triple (s, p, o), we want to make the embedding of o ($\boldsymbol{\phi}_o$) be the nearest neighbor of $\boldsymbol{\phi}_s + \boldsymbol{\phi}_p$ where the distance is measured by a dissimilarity function $d(\boldsymbol{\phi}_s + \boldsymbol{\phi}_p, \boldsymbol{\phi}_o)$ such as L_2-norm. Therefore, the distance score of a candidate o for given (s, p) can be measured as follows when L_2-norm is used as the dissimilarity function:

$$s'_{d_{spo}} = \sqrt{\sum_{j=1}^{m} (\phi_{s_j} + \phi_{p_j} - \phi_{o_j})^2} \tag{6.6}$$

where m denotes the dimensionality of the factorization/embedding for s, p, and o. Here we use L_2-norm as our dissimilarity function in the same way as the original study (Bordes et al., 2013). Afterwards, the candidate set of

objects can be ranked by their distance scores, where an *object* with a higher score should be ranked lower. The loss to be optimized in *TransE* can be defined as below in our settings (Bordes et al., 2013):

$$\ell(b_1, b_2) = \sum_{b_1 \in \mathcal{D}^+_{spo}} \sum_{b_2 \in \mathcal{D}^-_{spo}} [\gamma + s'_{b_1} - s'_{b_2}]_+ \tag{6.7}$$

where $\mathcal{D}^+_{spo}$ and $\mathcal{D}^-_{spo}$ denote the set of positive and negative training instances, respectively. Here, a positive instance denotes a valid triple (s, p, o), which can be found in the training set, and a negative instance consists of s, p, and a randomly chosen *object* o^- which does not exist in the training set. $[x]_+ = 0 \; for \; x < 0$, and x otherwise, and γ is a margin hyperparameter. In the same way as Bordes et al., 2013, we set γ to 1.0, and use L_2-norm as our dissimilarity function.

Transfer Learning via a Co-Factorization Model for the Two Tasks

As we can see from Equation 6.4 and 6.6, we have two related representations for the latent factors of an item i in the item recommendation task (or *subject* s in the KG completion task), i.e., $\boldsymbol{\theta}_i$ and $\boldsymbol{\phi}_s$, in the context of the two different tasks. In this work, we investigate two strategies for modeling the relationship between the two representations of items/subjects for transfer learning between the two tasks. For the sake of simplicity, we consider simple cases of $\mathbf{x}_{\mathbf{d_{ui}}}$, i.e., two categorical features (to denote u and i) for $\mathbf{x}_{\mathbf{d_{ui}}}$.

Shared latent space ($CoFM_A$). A straightforward approach to model the relationship between two representations for the latent factors of an item/subject in the two tasks is to assume that their latent factors are exactly the same, i.e., $\boldsymbol{\theta}_i = \boldsymbol{\phi}_s = \boldsymbol{\rho}_{is}$, where $\boldsymbol{\rho}_{is}$ is the same latent factor for both. Given this assumption, the preference score functions (Equation 6.4 and 6.6) for the aforementioned two tasks can then be re-written as:

$$s_{d_{ui}} = \beta_0 + \boldsymbol{\beta}_u + \boldsymbol{\beta}_i + < \boldsymbol{\theta}_u, \underline{\boldsymbol{\rho}_{is}} >, \quad s'_{d_{spo}} = \sqrt{\sum_{j=1}^{m} (\underline{\rho_{is_j}} + \phi_{p_j} - \phi_{o_j})^2} \tag{6.8}$$

This approach is based on a strong assumption that an item and a subject from the two different tasks have the same latent representation.

Via latent space regularization ($CoFM_R$). An alternative approach to work with the two latent representations of an item/subject is regularizing these representations to make them not reside too far away from each other. Therefore,

compared to the assumption in $CoFM_A$ that the two item/subject representations in the two different tasks are exactly the same, $CoFM_R$ allows the two representations to be different. However, the loss will be increased if the difference between the two representations is huge. We incorporate this intuition into the model by imposing the following regularization:

$$\lambda_{\boldsymbol{\phi},\boldsymbol{\theta}} \|\boldsymbol{\phi}_s - \boldsymbol{\theta}_i\|_F^2 \tag{6.9}$$

where $\lambda_{\boldsymbol{\phi},\boldsymbol{\theta}}$ is a regularization parameter.

Another issue for transfer learning between the two tasks is the different output scales of the two loss functions: Equation 6.5 and 6.7. Hence, we modify the loss function of the KG completion task (Equation 6.7) as follows in order to make both loss functions in the two tasks have the same scale.

$$\ell(b_1, b_2) = \sum_{b_1 \in \mathcal{D}^+_{spo}} \sum_{b_2 \in \mathcal{D}^-_{spo}} -\log[\delta(\gamma + s'_{b_1} - s'_{b_2})]_+ \tag{6.10}$$

Summary. Putting everything together, our co-factorization model in the view of *transfer learning* can be formulated as follows:

$$Opt(CoFM) : Opt(T) + \epsilon \times Opt(S), \tag{6.11}$$

$$Opt(T) = \arg\min \sum_{d_T \in \mathcal{D}_T} \ell_T(\cdot), \quad Opt(S) = \arg\min \sum_{d'_S \in \mathcal{D}'_S} \ell_S(\cdot) \tag{6.12}$$

where ϵ is a transfer (auxiliary) parameter to denote the importance of the knowledge transfer from the source task (S) to the target task (T). Let $\mathcal{D}_T$ and $\mathcal{D}_S$ denote the original training instances in the target and source tasks, respectively. $\mathcal{D}'_S$ is a set of training instances that is randomly sampled from $\mathcal{D}_S$ in order to match the size of $\mathcal{D}_T$, i.e., $|\mathcal{D}_T| = |\mathcal{D}'_S|$. For each instance $d_T \in \mathcal{D}_T$, we choose an instance d'_S randomly with a replacement from $\mathcal{D}_S$ where the *item* in d_T is the same as the *subject* in d'_S, i.e., $d_T(i) = d'_S(s)$. With the same size for both T and S, we then use the SGD to learn the parameters in the *CoFM*.

An overview of the algorithm to optimize Equation 6.11 using SGD is presented in Algorithm 3 when the target task is item recommendations. Our approach can be seen as a *transfer learning* (Pan and Yang, 2010) model as we are transferring knowledge between two different but related tasks in the same domain. It is worth noting that, in contrast to *multi-task learning* which aims to learn both tasks simultaneously, *transfer learning* aims to achieve the best performance for T with the transferred knowledge from S.

Algorithm 3: Main elements of the algorithm to optimize Eq. 6.11 using SGD when the target task T is item recommendations.

input : training datasets $\mathcal{D}_{ui}$ and $\mathcal{D}'_{spo}$ with the same size of $|\mathcal{D}|$, initialized parameters for CoFM

output: learned parameters for CoFM

repeat
  for d_{ui} *in* $\mathcal{D}_{ui}$ **do**
    Optimize Opt(T) for $\theta_u, \theta_i, \beta$
      perform SGD for BPR loss function in terms of d_{ui}
    Select d'_{spo} **in** $\mathcal{D}'_{spo}$ **where** $d'_{spo}(s) = d_{ui}(i)$
    Optimize Opt(S) for ϕ_s, ϕ_p, ϕ_o
      perform SGD for BPR loss function in terms of d'_{spo}
until *converged*;

6.3.2 Datasets

We use two datasets in the movie and book domains, which have been widely used in previous studies with respect to LODRecSys (Noia et al., 2016; Piao and Breslin, 2017a; Musto et al., 2016b).

- **Movielens dataset** (Noia et al., 2016). This dataset is the one used in the previous section. It consists of users and their ratings about movies, and each of the items in this dataset has been mapped to a DBpedia entity if there is a mapping available. In the same way as previous studies (Noia et al., 2016; Piao and Breslin, 2017a), we consider ratings higher than 3 as positive feedback and others as negative ones.
- **DBbook dataset**. The dataset[6] consists of users and their binary feedback (1 for likes, and 0 otherwise), where the items have been mapped to DBpedia entities if there is a mapping available.

Table 6.5 shows the main details of user-item interactions and RDF triples associated with items in the two datasets. There are 3,997 users and 3,082 items with 827,042 ratings in the Movielens dataset. The DBbook dataset consists of 6,181 users and 6,733 items with 72,372 interactions. The sparsity of the DBbook dataset (99.38%) is higher than that of the Movielens dataset (93.27%). For item recommendations, we use 80% and 20% of each dataset for training and test sets. 20% of the training set was used for tuning hyperparameters, and a model was re-trained using the whole training set later. In addition, all of the items were considered as candidate items

[6] http://challenges.2014.eswc-conferences.org

TABLE 6.5: Statistics of Movielens and DBbook datasets.

		Movielens	**DBbook**
statistics of user-item interactions	# of users	3,997	6,181
	# of items	3,082	6,733
	# of ratings	827,042	72,372
	avg. # of ratings	206	12
	sparsity	93.27%	99.38%
	% of positive ratings	56%	45.85%
statistics of RDF triples	# of subjects	2,952 (3,082)	6,211 (6,733)
	# of predicates	21	36
	# of objects	18,550	16,476
	# of triples	81,835	72,911

for recommendations in the same way as in Noia et al., 2016 instead of considering only "rated test-one" evaluation.

The second part of Table 6.5 shows the details of extracted triples for items/subjects in the two datasets from the DBpedia SPARQL endpoint. In the Movielens dataset, 2,952 out of 3,082 (95.8%) items have at least one triple. There are 21 distinct predicates and 18,550 objects in the Movielens dataset, which results in 81,835 triples in total. In the case of the DBbook dataset, 6,211 out of 6,733 (92.2%) items have at least one triple. There are 36 distinct predicates and 16,476 objects in the DBbook dataset, which results in 72,911 triples in total. For KG completion with respect to the domain of items, we adopt the same splitting strategy as Drumond et al., 2012 for constructing training and test sets. We randomly choose a *subject* and *predicate* pair (s, p) for a given s, and then use all triples containing the pair to construct the test set. The other triples with the same *subject* were put into the training set.

We repeated five times by sampling new training and test sets for the two tasks using the aforementioned strategies, and applied different methods to them. The results in Section 6.4 are based on the averages over five runs.

6.3.3 Compared methods

We use $CoFM_A$ to denote the *CoFM* method which shares latent space with the assumption that two latent factors of an item/subject in the two tasks are exactly the same, and use $CoFM_R$ to denote the *CoFM* method which uses

regularization for modeling the relationship between the two latent factors of an item/subject in the two tasks.

Parameter settings of CoFM. The transfer (auxiliary) parameter ϵ was determined by a separate validation set randomly retrieved from 20% of the training set for the first run in terms of the loss on the target task in each dataset. According to the results, ϵ was set to 0.05 for the Movielens dataset when either KG completion or item recommendations is the target task. For the DBbook dataset, ϵ was set to 0.05 and 1.0 when KG completion and item recommendations is the target task, respectively. In addition, we set the same value for all regularization parameters in our approach for the sake of simplicity. $\lambda = 0.01$ when item recommendations is the target task, and $\lambda = 0.001$ when KG completion is the target one. The dimensionality value m was set to 64, which is the same as in Drumond et al., 2012, for all factorization-based approaches.

We compare *CoFM* against the following methods for item recommendations.

- *kNN-item (kNN): kNN-item* is an item-based k-nearest neighbors algorithm. We use a MyMedialite[7] implementation for this baseline where $k = 80$.
- *BPRMF* (Rendle et al., 2009): *BPRMF* is a matrix factorization approach for learning latent factors with respect to users and items, optimized for BPR. *BPRMF* can be seen as the model for item recommendations in *CoFM*, which is a FM model without transferring knowledge from the KG completion task.
- *LODFM* (Piao and Breslin, 2017a): This is the approach we presented in the previous section. *LODFM* exploits *lightweight* KG-enabled features about items from DBpedia, which can be obtained directly from its SPARQL endpoint.

For the KG completion task, we compare *CoFM* against the following methods:

- *MFPP:* Most Frequent Per Predicate (*MFPP*) is a baseline method which recommends the *objects* that co-occur most frequently with the *predicate* p given a *subject* s and *predicate* pair (s, p).
- *PITF* (Rendle and Schmidt-Thieme, 2010): This model has been proposed in Rendle and Schmidt-Thieme, 2010 for tag recommendations. In Drumond et al., 2012, the authors applied a *PITF* model optimized for the BPR criterion, which captures the interactions among *subjects*,

[7] http://www.mymedialite.net/

predicates, and *objects* of RDF triples. We re-implement this approach under the framework of FMs.

- *TransE* (Bordes et al., 2013): This is a *translation-based* approach which models relationships by interpreting them as translations operating on the entity embeddings. We re-implemented this approach based on the parameters from Bordes et al., 2013. As one might expect, *TransE* can be seen as the model for the KG completion task in *CoFM* without transferring knowledge from item recommendations.

6.4 Results

Table 6.6 and 6.7 show the results of comparing $CoFM$ with the aforementioned methods in each task on the Movielens and DBbook datasets. Overall, $CoFM$ provides the best performance compared to other approaches in terms of item recommendations as well as KG completion in both datasets.

As we can see from Table 6.6, $CoFM_R$ provides the best performance, and improves the recommendation performance significantly ($p < 0.01$) compared to kNN and $BPRMF$ for item recommendations on the Movielens dataset. Similarly, $CoFM_R$ outperforms baselines such as $MFPP$ and $PITF$ significantly for KG completion. In detail, a significant improvement of $CoFM_R$ over $PITF$ in MRR (+21%), nDCG@5 (+19.8%), P@5 (+31.2%), and R@5 (+8.2%) can be noticed.

On the DBbook dataset (Table 6.7), $CoFM_A$ provides the best performance instead of $CoFM_R$. $CoFM_A$ outperforms kNN and $BPRMF$ significantly for item recommendations, and outperforms $MFPP$ and $PITF$ for the KG completion task ($p < 0.01$). One of the possible explanations for the observation that the best performance is achieved by $CoFM_R$ for the Movielens dataset and by $CoFM_A$ for the DBbook one might be due to the different sparsity levels of the two datasets. As we can see from Table 6.5, the DBbook dataset has higher sparsities compared to the Movielens dataset for both tasks. $CoFM_A$, which can be seen as having strong knowledge transfer with the assumption that item/subject embeddings in the two tasks are the same, may possibly be more useful for this sparse dataset and leads to better performance.

$LODFM$ vs. $CoFM$. We observe that $CoFM_R$, which incorporates the *incompleteness* of DBpedia, outperforms $LODFM$ which leverages *existing* knowledge from DBpedia on the Movielens dataset. A significant difference between the two approaches in terms of all evaluation metrics can be noticed ($p < 0.01$). On the DBbook dataset, $CoFM_A$ also consistently outperforms

TABLE 6.6: Results of *KG completion* and *item recommendations* on the Movielens dataset. *S* denotes source task while *T* denotes target task. The gray cells denote significant improvement over the best-performing baseline.

(A) *S*: KG completion, *T*: item recommendations

	kNN	*BPRMF*	*LODFM*	$CoFM_A$	$CoFM_R$
MRR	0.510	0.594	0.609	0.602	**0.622**
nDCG@5	0.358	0.425	0.436	0.429	**0.445**
P@5	0.291	0.355	0.366	0.360	**0.372**
R@5	0.075	0.097	0.100	0.098	**0.102**
nDCG@10	0.440	0.500	0.510	0.504	**0.518**
P@10	0.258	0.307	0.314	0.310	**0.318**
R@10	0.129	0.161	0.165	0.164	**0.170**
nDCG@20	0.583	0.645	0.653	0.648	**0.660**
P@20	0.218	0.252	0.257	0.254	**0.259**
R@20	0.213	0.257	0.261	0.260	**0.265**

(B) *S*: item recommendations, *T*: KG completion

	MFPP	*PITF*	*TransE*	$CoFM_A$	$CoFM_R$
MRR	0.183	0.266	0.317	0.302	**0.322**
nDCG@5	0.149	0.248	0.292	0.279	**0.297**
P@5	0.070	0.096	0.123	**0.126**	**0.126**
R@5	0.103	0.230	0.241	0.240	**0.249**
nDCG@10	0.171	0.273	0.311	0.299	**0.316**
P@10	0.046	0.064	0.077	**0.081**	0.079
R@10	0.149	0.271	0.277	0.280	**0.283**
nDCG@20	0.194	0.297	0.331	0.321	**0.336**
P@20	0.031	0.042	0.047	**0.051**	0.048
R@20	0.199	0.311	0.313	0.318	**0.319**

LODFM in terms of all evaluation metrics, and specifically in terms of precision, e.g., +8.3% of P@5, +8% of P@10, and +5.6% of P@20 ($p < 0.01$). The results show that incorporating the incompleteness of the KG improves the performance of item recommendations significantly.

With vs. Without knowledge transfer. We now look at the results of *CoFM* with and without transferring knowledge between the two tasks. *BPRMF* and *TransE* can be seen as the *CoFM* without transferring knowledge between these tasks. On the Movielens dataset, $CoFM_R$ improves the performance by 2.3%-5.2% compared to *BPRMF* for item recommendations ($p < 0.01$). Regarding the KG completion task, $CoFM_R$ outperforms *TransE* significantly for all evaluation metrics as well. On the DBbook dataset. $CoFM_A$ improves

TABLE 6.7: Results of *KG completion* and *item recommendations* on the DBbook dataset. S denotes source task while T denotes target task. The gray cells denote significant improvement over the best-performing baseline.

(A) S: KG completion, T: item recommendations

	kNN	*BPRMF*	*LODFM*	$CoFM_A$	$CoFM_R$
MRR	0.015	0.115	0.121	**0.125**	0.100
nDCG@5	0.008	0.105	0.110	**0.114**	0.091
P@5	0.003	0.034	0.036	**0.039**	0.031
R@5	0.010	0.096	0.101	**0.106**	0.085
nDCG@10	0.014	0.125	0.131	**0.134**	0.108
P@10	0.004	0.024	0.025	**0.027**	0.022
R@10	0.023	0.135	0.141	**0.147**	0.116
nDCG@20	0.022	0.145	0.153	**0.156**	0.126
P@20	0.004	0.017	0.018	**0.019**	0.015
R@20	0.043	0.187	0.196	**0.198**	0.138

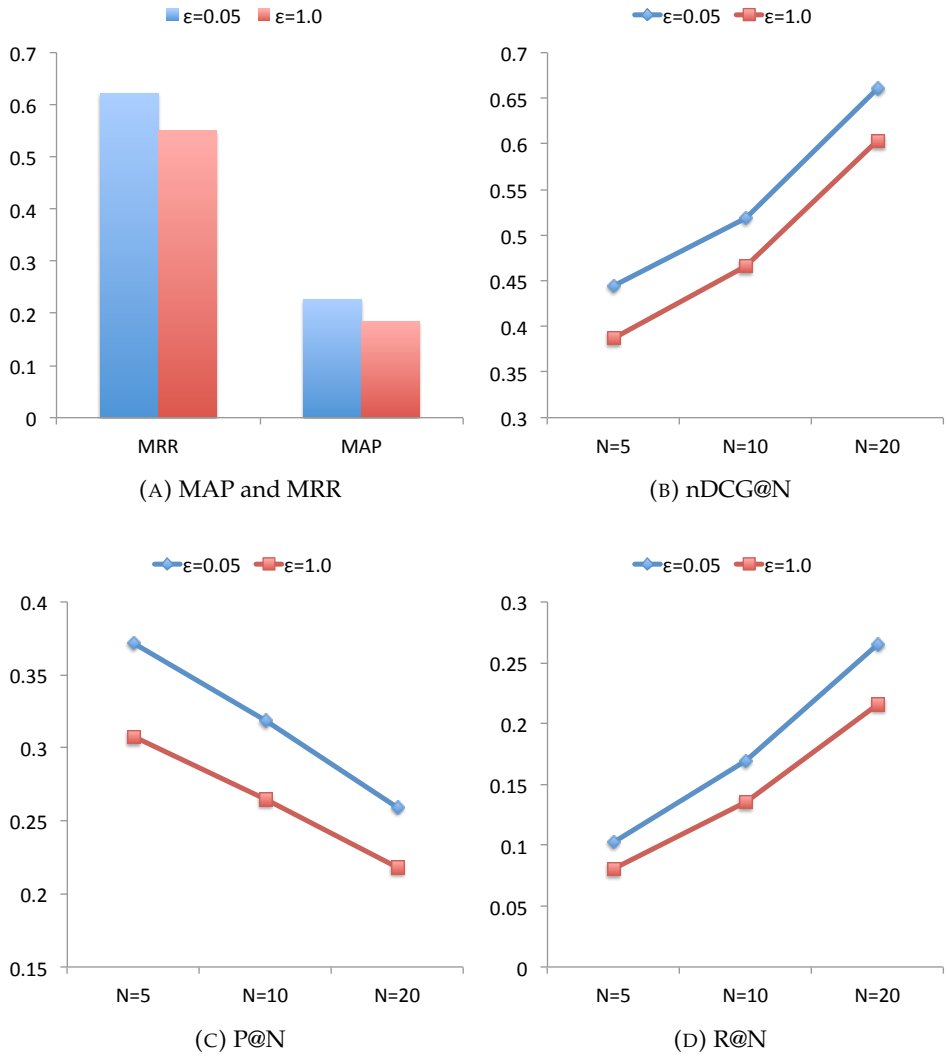
(B) S: item recommendations, T: KG completion

	MFPP	*PITF*	*TransE*	$CoFM_A$	$CoFM_R$
MRR	0.168	0.383	0.408	**0.412**	**0.412**
nDCG@5	0.162	0.372	0.399	**0.410**	0.400
P@5	0.048	0.111	0.117	**0.119**	**0.119**
R@5	0.177	0.363	0.377	0.380	**0.381**
nDCG@10	0.181	0.389	0.416	**0.423**	0.416
P@10	0.031	0.064	0.067	**0.068**	0.067
R@10	0.220	0.396	0.406	**0.408**	**0.408**
nDCG@20	0.203	0.404	0.428	**0.434**	0.430
P@20	0.021	0.037	0.037	**0.038**	**0.038**
R@20	0.279	0.428	0.433	0.435	**0.436**

the performance by 5.9%-14.7% compared to *BPRMF* for item recommendations. For the KG completion task, $CoFM_A$ outperforms *TransE* significantly in terms of all evaluation metrics except R@10. This indicates that transferring knowledge between the two tasks improves the performance on both tasks compared to each single model without transferring knowledge from the other task.

With vs. Without tuning the transfer parameter ϵ. Figure 6.7 shows the results of item recommendations on the Movielens dataset using $CoFM_R$ with a tuned value for the parameter ϵ ($\epsilon = 0.05$) and without tuning the parameter

($\epsilon = 1.0$). As we can see from the figure, tuning the transfer value ϵ plays an important role in achieving the best performance for the target task.



(A) MAP and MRR

(B) nDCG@N

(C) P@N

(D) R@N

FIGURE 6.7: The performance of item recommendations on the Movielens dataset with $\epsilon = 0.05$ and $\epsilon = 1.0$ using $CoFM_R$.

To sum up, the implications of these results are twofold. With a proper transfer parameter, (1) incorporating the *incompleteness* of a KG can improve the performance of item recommendations, and (2) the knowledge from item recommendations, i.e., *user-item interaction histories* can also be transferred to the task of KG completion with respect to the domain of items, which improves the performance significantly.

6.5 Summary

In this chapter, we proposed *bottom-up* approaches to learn the latent representations of users and items to provide personalized recommendations for well-established users.

In Section 6.2, we investigated using a FM with lightweight LOD-enabled features, such as the predicate-object lists, subject-predicate lists, and PageRank scores of items which can be directly obtained from the DBpedia SPARQL endpoint, for top-N recommendations in the movie and music domains. The results show that our proposed approach significantly outperforms compared approaches such as *SPRank* and *BPRMF*.

In addition, we analyzed the recommendation performance based on different combinations of features. The results indicate that using the predicate-object list and the PageRank scores of items can provide the best performance. On the other hand, including the subject-predicate list of items is not helpful in improving the quality of recommendations but rather decreases the performance.

In Section 6.3, we investigated transfer learning between item recommendations and knowledge graph completion with a co-factorization model. The intuition behind transfer learning between the two tasks is (1) to incorporate the incompleteness of a KG for item recommendations, and (2) to investigate whether the knowledge from item recommendations can also be transferred to the KG completion task and improves its performance.

The experimental results are promising and suggest that incorporating the incompleteness of a KG improves the recommendation performance significantly compared to *LODFM*, which uses existing knowledge from a KG, and outperforms other baselines as well. In addition, we further explored potential synergies that transfer knowledge from item recommendations, i.e., *user-item interaction histories* to the task of KG completion, which has not been explored in previous studies. Results indicate that the knowledge from *user-item interaction histories* can be transferred to the KG completion task, and improves its performance significantly.

Chapter 7

Conclusions and Future Work

Online social networking platforms have been embedded into our daily lives. Nowadays billions of people are using OSNs every day around the world, and the number keeps growing with an estimation of 2.44 billion people using these platforms by 2018[1]. The ever-increasing data generated by users in those OSNs poses new challenges as well as opportunities for inferring their interests in order to provide personalized services such as recommendations. In this thesis, we addressed the challenges of inferring user interest profiles for recommender systems in OSNs in the context of different cold-start scenarios based on proposed semantics-aware approaches.

7.1 Summary of Contributions

The main findings and contributions of this thesis with respect to the research challenges identified in Section 2.5 can be summarized as follows.

Semantics-aware user modeling on microblogging social networks

Knowledge graphs such as DBpedia go beyond just categories to provide related entities via the entity's predicates, which is the key difference between other knowledge bases such as Wikipedia. Also, various types of information about users' followees such as their *biographies* and *list memberships* have great potential for inferring user interest profiles for passive users. Therefore, it is important to leverage different aspects of KGs and various types of user activities in OSNs for inferring user interest profiles. To this end, we investigated the following research questions.

- How can we leverage knowledge graphs to infer and represent user interest profiles from different types of user activities on microblogging social networks?

[1] https://goo.gl/axSrzS

- How can we incorporate different user modeling dimensions such as the temporal dynamics of user interests in order to construct better user interest profiles?

With respect to the first research question, we investigated various interest propagation strategies that leverage different aspects of DBpedia such as *category-*, *class-*, and *predicate-based* approaches for enhancing the primitive interests of users (Section 3.5). The results show that the *category* & *predicate-based* propagation strategy outperforms other core extension strategies as well as other combined strategies. In addition, we proposed a hybrid representation strategy which combines WordNet synsets and DBpedia concepts for constructing user interest profiles in Section 3.7. This approach aims at overcoming the limitations of representing user interest profiles using DBpedia concepts alone, which might miss emerging user interests (entities) that have not existed in the KG.

Furthermore, we investigated different types of information about users' followees such as their *biographies* (Section 4.2) and *list memberships* (Section 4.3) in order to infer user interest profiles for *passive* users who do not post content but who keep following other people for their information needs. We showed that user interest profiles constructed based on these two types of information about followees perform better compared to the ones based on the tweets of followees and the topical-followees approach. Also, the experimental results in Section 4.4 indicate that the information from *biographies* and *list memberships* of followees complements each other and can improve the user modeling performance.

To answer the second question, in Section 3.7, we investigated the quality of user interest profiles by combining the best-performing strategy in four user modeling dimensions such as *Interest Representation*, *Content Enrichment*, *Temporal Dynamics*, and *Interest Propagation* in our proposed user modeling framework. Based on the optional components of these four user modeling dimensions, we compared the URL recommendation performance using 16 user interest profiles generated by all possible options. The experimental results show that *Interest Representation* and *Content Enrichment* play crucial roles in user modeling, followed by *Temporal Dynamics*. In contrast, although propagating user interests leveraging the background knowledge from DBpedia improves the performance when we use concept-based user interest profiles, the *Interest Propagation* dimension had little effect on user modeling when considering different dimensions together, e.g., with enriched content or rich representation strategies.

Similar findings have been observed recently in Manrique and Mariño, 2017 in the context of recommending research papers. In Manrique and

Mariño, 2017, the authors showed that propagating user interests based on the background knowledge from DBpedia improves the recommendation performance when only abstracts are available, but has little effect when the full-texts of papers (rich information) are also available.

Semantic similarity measures for recommending items in cold-start scenarios

Semantic similarity/distance measures such as *LDSD* which measure the similarity between two entities in a KG are useful for providing recommendations for new users who have little or no explicit feedback with respect to items. However, KGs are far from complete Galárraga et al., 2017 despite the fact that they provide billions of machine-readable facts about entities, and it is crucial to understand the effect of the incompleteness of KGs on recommender systems based on a semantic similarity/distance measure. In this regard, we investigated the following research questions in this thesis.

- How can we improve the performance of *LDSD* by resolving some limitations of it?
- Do different sparsities of background knowledge from KGs with respect to items affect the performance of recommendations based on semantic similarity/distance measures?

For the first research question, we proposed a semantic distance measure called *mLDSD* in **Chapter 5**, and evaluated our proposed approach against other semantic similarity/distance measures in the context of recommending items in the music domain, and showed that the approach outperforms other semantic similarity/distance measures such as *LDSD* significantly. We investigated several normalization strategies for *mLDSD*, and showed that a global normalization strategy which penalizes the importance of a path between two entities according to the global appearances of the path in the whole DBpedia graph performs best compared to other normalization strategies.

Regarding the second question, we investigated the *Linked Data sparsity problem* which denotes that the performance of the recommender system based on semantic similarity/distance measures decreases when entities lack information (i.e., when they have a small number of incoming/outgoing relationships to other entities). Through the experiment conducted in Section 5.6, we showed that the recommendation performance based on those semantic similarity/distance measures has a very strong positive

relationship with the number (log scale) of the total number of incoming/outgoing links ($p < 0.01$) for entities.

Semantics-aware machine learning approaches for item recommendations

With respect to Linked Open Data-enabled recommender systems, leveraging LOD-enabled features requires additional steps in addition to well-established recommendation approaches such as (1) retrieving background knowledge from KGs, (2) building and maintaining a combined graph based on user-item interactions and the background knowledge about items, (3) extracting useful features from the graph built, and (4) feeding them to various recommendation approaches. The complicated process of consuming LOD for RS will hinder the adoption of LOD for the RS community, and this has in turn motivated us to investigate the following research question:

- How can we ease the process of leveraging background knowledge from KGs for item recommendations while having competitive performance compared to previous semantics-aware approaches?

To address this research question, we proposed *LODFM* which leverages lightweight LOD-enabled features using FMs in Section 6.2. Differing from most approaches for LODRecSys, *LODFM* directly consumes lightweight LOD-enabled features which are queried from a SPARQL endpoint of KGs such as DBpedia. The results show that *LODFM* can also achieve state-of-the-art performance compared to other baselines.

Previous studies including *LODFM* have focused on exploiting the existing knowledge about items in KGs, and have not considered the incompleteness of a KG. In addition, whether the knowledge can be transferred from the other direction, i.e., from the item recommendation task to the KG completion one, has not been explored. We filled this research gap by addressing the question below.

- Does transfer learning between the two tasks improve the performance compared to the approaches without transfer learning for each task?

To answer the research question, we investigated transfer learning between the two tasks: (1) item recommendations, and (2) knowledge graph completion with respect to the item related domain with a co-factorization model. Through the experimental results, we showed that incorporating the incompleteness of a KG via transfer learning between the two tasks can improve the performance of item recommendations. The results also indicate that transferring knowledge from item recommendations to the KG completion

task can also improve the performance of KG completion which has not been shown in previous studies.

In conclusion, this thesis contributes to research on inferring user interest profiles in microblogging OSNs as well as recommender systems in OSNs in the context of different scenarios. First, we proposed several different user modeling strategies to infer user interest profiles for either *active* or *passive* users. These user modeling strategies explore different aspects of DBpedia and various types of activities of users in OSNs, and also incorporate various user modeling dimensions. Secondly, we introduced *mLDSD* and *LODFM*, which explore the background knowledge of items from DBpedia items to provide recommendations in different situations such as a cold start. Finally, we investigated knowledge transfer between item recommendations and knowledge graph completion in order to incorporate the incompleteness of a KG, and examined whether the knowledge from item recommendations can also be transferred to the KG completion task and can improve its performance.

7.2 Discussions

Every rose has its thorn, and the thesis also has some limitations which we will discuss in the following.

The active users in this thesis are defined as the ones who have posted more than 100 tweets (see Section 3.1). This does not consider the time distribution of tweets. For example, users tweeted consistently during each week in their historical UGC and the ones tweeted during a certain period and stoped are all considered as active users in our study. An alternative definition of active users can be the ones who consistently posted more than h tweets during each week where h is a threshold.

For evaluating inferred user interest profiles in the context of URL recommendations, we filtered topical URLs which have at least four concepts (see Section 3.4.1). This limited the evaluation of different user modeling strategies to recommending URLs having longer content. Compared to other previous studies (Abel et al., 2011c; Abel et al., 2013b) which evaluate different user modeling strategies in terms of news recommendations, the topical URLs that we filtered covered different types of contents such as blog posts and websites. However, these user modeling strategies might not work well for recommending items that have short content (e.g., tweet recommendations).

In Chapter 4, we proposed leveraging the biographies and list memberships of followees for inferring user interests of passive users. We simulated passive users with those active users used in Chapter 3 by blinding out their tweets in order to have ground truth URLs for evaluation. As the number of followees for passive users can be different for active ones, we considered different numbers of followees (50, 100, 150, and 200) for passive users in the experiment (see Section 4.3.4). In Gong et al., 2015, the authors showed that passive users have 190 and 266 followees on average in Singapore and Indonesia Twitter communities. This shows that passive users still follow many users on Twitter which can be used for inferring their interest profiles, and also indicates that our simulation is valid.

An alternative way of evaluating inferred user interest profiles for passive users can be constructing ground truth from other platforms where those users have left their interest history. For example, passive Twitter users can have their music preferences on Spotify[2] or their movie preferences on IMDB[3]. A recent work by Tommaso et al., 2018 provides a user interests dataset which includes an average of 90 multi-domain preferences per user on music, books, movies, etc.

Another limitation of evaluating inferred user interest profiles based on the bios or list memberships of followees in the context of URL recommendations is the absence of the information when the followeeships were made. For example, we cannot guarantee that a user shared an URL after following certain accounts, and the user might have tweeted that URL in the first place.

Although we focused on user modeling strategies for passive users in Chapter 4, these strategies leveraging the biographies and list memberships of followees for inferring user interest profiles can be applied to active users as well. In Besel et al., 2016b, the authors showed that the cosine similarity between user interest profiles based on entities extracted from their tweets and the ones based on entities extracted from followees accounts is 0.66. This result suggests that the information from the tweets of users and their followees can be complement each other to provide a comprehensive user interest profiles.

All those proposed user modeling strategies in Chapters 3 and 4 have been evaluated in terms of the accuracy of URL recommendations. However, other evaluation metrics have been proposed in the recommender systems community as a recommender system with a high accuracy is not enough to provide good recommendations. For example, evaluation metrics such as *diversity* and *serendipity* aim to measure the quality of recommendations in

[2] https://www.spotify.com
[3] https://www.imdb.com/

terms of the diversity of recommended items and whether the recommended items are surprising to users (Kaminskas and Bridge, 2016). Our proposed user modeling strategies leverage background knowledge for propagating user interests, and those propagated interests based on the background knowledge can lead to diverse and surprise topics. For example, Figure 7.1 shows some related categories with respect to the entity `dbr:IPad`. As we can see from the figure, the propagated interests (categories) cover diverse topics and some of them can be used to retrieve suprising URLs (e.g., retrieving news articles related to `dbc:Foxconn`). However, these assumptions should be tested carefully with the consideration of the trade-off between the diversity (or serendipity) and the accuracy of recommended items.

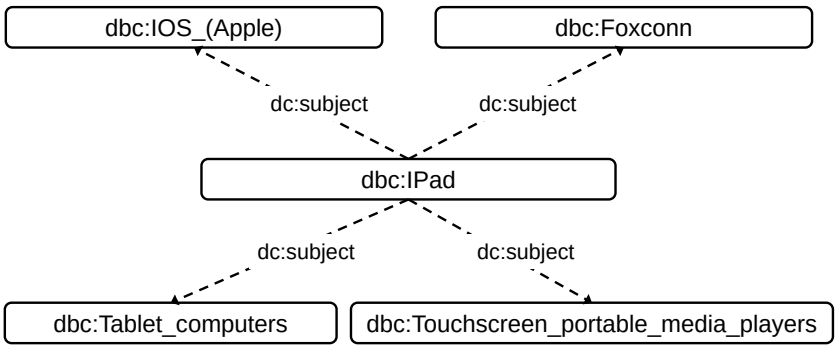


FIGURE 7.1: Pieces of information about the entity `dbr:IPad` from DBpedia.

In Chapter 6, we used a negative sampling approach BPR (Bayesian Personalized Ranking) for learning parameters in factorization models. The assumption behind this approach is that a liked item for a user should be ranked higher (with a higher score) compared to a random one in the list of items with which the user has not interacted. Despite the fact that negative sampling approaches such as BPR has been widely used for learning parameters in many collaborative filtering approaches (Hong et al., 2013; Rendle, 2012), the assumption is not perfect as a user might like some of the randomly chosen items (negative samples) even the user has not interacted with those items.

In addition, we evaluated our proposed factorization models such as *LODFM* and *CoFM* using two datasets in different domains such as the

movie and music domains. Therefore, our proposed models are not limited to a specific domain and can be applied to item recommendations in other domains with background knowledge about those domain-specific items.

7.3 Future Work

On top of the proposed approaches and findings with respect to semantics-aware user modeling and recommender systems in OSNs in previous chapters, in this section we outline some research directions that are worth exploring in the future in the areas of user modeling and recommender systems in the domain of OSNs.

First, more sophisticated approaches for understanding the semantics of UGC are required. For example, for those approaches that rely on extracted entities for inferring user interest profiles, extracting entities from microblogs is a fundamental step which is challenging by itself. The uncertainty (confidence) of the extracted entities can be incorporated into the weighting scheme for the primitive interests of users as well as the enhanced ones. Moreover, most approaches have extracted explicitly mentioned entities based on NLP APIs such as tag.me, Aylien, OpenCalais, etc. However, there can be many entities implicitly mentioned in tweets. In Perera et al., 2016, the authors showed that over 20% of mentions of movies are implicit references, e.g., a tweet referring the movie *Gravity* - "ISRO sends probe to Mars for less money than it takes Hollywood to make a movie about it". It shows that advanced methods for extracting entities, such as the one proposed in Perera et al., 2016, have great potential to improve the quality of user modeling. Also, considering the context of a microblog might be useful when extracting entities instead of just considering the single microblog of a user. The context might refer to some previous microblogs posted by the user, or other microblogs with the same hashtag in the microblogging service.

Furthermore, polyrepresentation of user interests can be further studied. It is not necessary to maintain several user interest profiles for a single user, but a single model can also be built with relevant information from different aspects, and a view/aspect made for the user based on the information needs for different applications. GeniUS (Gao et al., 2012) is a good example in this regard, which is a user modeling library that stores concept-based user interest profiles using the RDF format (a W3C recommendation) with widely used ontologies such as FOAF (Brickley and Miller, 2012), SIOC (Breslin et al., 2005), and WI[4]. In GeniUS, user interest profiles are represented as

[4] http://smiy.sourceforge.net/wi/spec/weightedinterests.html

DBpedia entities and enriched by background knowledge such as the type (domain) of an entity from DBpedia. Therefore, the constructed profile is flexible enough to retrieve its sub-profiles with respect to specific domains (e.g., `Music`), which is useful for recommending domain-specific items. The idea is that, for example, one only needs your music-related interest profile in the context of music recommendations. The results in Gao et al., 2012 indicate that domain-specific profiles clearly outperform the whole user profiles for domain-specific tweet recommendations in terms of six different domains. Although GeniUS only considers different views of users in terms of topical domains, the same idea can be extended to other views. For instance, different user profiles can be extracted dynamically with different approaches for incorporating temporal dynamics, e.g., retrieving short-term profiles for recommending tweets during an event, which might be more useful compared to using long-term profiles.

Secondly, although many *distance-based* approaches have been proposed for measuring the semantic similarity/distance between two entities in a linked dataset such as DBpedia, there exists little work on *feature-based* semantic similarity measures (Meymandpour and Davis, 2016). In contrast to distance-based measures, feature-based ones have two merits: (1) Given feature vectors of two entities, we can leverage well-established similarity measures such as the cosine similarity, *BM25*, and *BM25+* (Lv and Zhai, 2011) instead of "reinventing the wheel". (2) As we can extract features for each entity and build its feature vector beforehand, it is faster to compute the similarities based on those feature vectors compared to distance-based similarity measures which need to explore the paths between two entities.

Thirdly, the promising results with transfer learning between item recommendations and knowledge graph completion motivate us to investigate more sophisticated transfer learning approaches for both tasks. The approaches via shared representation and incorporating regularization form into the objective functions in both tasks in Section 6.3 have some limitations. For example, the latent representations of items/subjects in the two tasks have to have the same dimensionality in both approaches, which might not be necessarily true. As a further step, an investigation of other ways to model the relationships between two representations of an item/subject in the two tasks can be conducted, e.g., using different dimensions for representing items and subjects and modeling the transition relationship between those dimensions.

Another interesting research direction might be the extraction of background knowledge for items, such as the recent study from Lalithsena et al., 2016, which aims to extract a subgraph for domain specific recommendation systems. Most previous studies for LODRecSys have used the predicates

directly related to the items when those items are either subjects or objects. However, other predicates which indirectly related to items might be useful for providing item recommendations as well. For example, two movies might be similar due to the fact that both directors for these movies have won an Academy Award (Lalithsena et al., 2016).

Finally, other evaluation metrics beyond the accuracy of ranking, such as diversity, serendipity, novelty, and coverage (Kaminskas and Bridge, 2016), should be further studied for different user modeling strategies and recommendation approaches.

Appendix A

Other Activities During PhD

Challenges

- **1st** place in the Semantic Sentiment Analysis Challenge at the 15th Extended Semantic Web Conference (**ESWC**), Crete, Greece, 02/06/2018
- **4th** place in the Data Challenge on "Entity Type Prediction over Linked Data" (among 13 teams) at the 5th Joint International Semantic Technology Conference (**JIST 2015**), Yichang, China, 12/11/2015
- **Finalist** at the 31st ACM SAC Student Research Competition (sponsored by Microsoft), Pisa, Italy, 04/04/2016

Program Committee/Reviewer

- 2018, International Semantic Web Conference (**ISWC**)
- 2017, Insight Student Conference (PC and session chair)
- 2017, International Semantic Web Conference (**ISWC**)
- 2017, European Semantic Web Conference (**ESWC**)
- 2016, International Workshop on Educational Recommender Systems (**EdRecSys**) at Web Intelligence (**WI**)

Other Publications

- **G. Piao**, J. G. Breslin. DBQuote: A Social Web based System for Collecting and Sharing Wisdom Quotes [Poster]. The 5th Joint International Semantic Technology Conference (**JIST 2015**), Yichang, China, 2015 - (Piao and Breslin, 2015)

- **G. Piao**, J. G. Breslin. Analyzing MOOC Entries of Professionals on LinkedIn for User Modeling and Personalized MOOC Recommendations [Abstract]. The 24th Conference on User Modeling, Adaptation and Personalization (**UMAP 2016**), Halifax, Canada, 2016 - (Piao and Breslin, 2016e)

- **G. Piao**, J. G. Breslin. Financial Aspect and Sentiment Predictions with Deep Neural Networks: An Ensemble Approach [Workshop]. Financial Opinion Mining and Question Answering Workshop at The Web Conference (**WWW**), Lyon, France, 2018 - (Piao and Breslin, 2018d)

- **G. Piao**, J. G. Breslin. Learning to Rank Tweets with Author-based Long Short-Term Memory Networks. The 18th International Conference on Web Engineering (**ICWE**), Caceres, Spain, 2018 - (Piao and Breslin, 2018a)

- **G. Piao**, J. G. Breslin. A Study of the Similarities of Entity Embeddings Learned from Different Aspects of a Knowledge Base for Item Recommendations. The 1st Workshop on Deep Learning for Knowledge Graphs and Semantic Technologies at the 15th Extended Semantic Web Conference (**ESWC**), Crete, Greece, 2018 - (Piao and Breslin, 2018b)

- **G. Piao**, J. G. Breslin. Domain-Aware Sentiment Classification with GRUs and CNNs. **1st** place in the Semantic Sentiment Analysis Challenge at the 15th Extended Semantic Web Conference (**ESWC**), Crete, Greece, 2018 - (Piao and Breslin, 2018c)

- **G. Piao**, J. G. Breslin. Inferring User Interests in Microblogging Social Networks: A Survey - (Piao and Breslin, 2018e)

Bibliography

Abdel-Hafez, Ahmad and Yue Xu (2013). "A survey of user modelling in social media websites". In: *Computer and Information Science* 6.4, pp. 59–71. ISSN: 1913-8997.

Abel, Fabian (2011). "Contextualization, user modeling and personalization in the Social Web–from social tagging via context to cross-system user modeling and personalization". PhD thesis. Leibniz University of Hanover.

Abel, Fabian, Qi Gao, Geert-Jan Houben, and Ke Tao (2011a). "Analyzing temporal dynamics in Twitter profiles for personalized recommendations in the Social Web". In: *Proceedings of the 3rd International Web Science Conference*. Koblenz, Germany: ACM, pp. 1–8. ISBN: 1450308554.

Abel, Fabian, Qi Gao, Geert-Jan Houben, and Ke Tao (2011b). "Analyzing user modeling on Twitter for personalized news recommendations". In: *User Modeling, Adaption and Personalization*. Girona, Spain: Springer, pp. 1–12. ISBN: 3642223613.

Abel, Fabian, Qi Gao, Geert-Jan Houben, and Ke Tao (2011c). "Semantic enrichment of Twitter posts for user profile construction on the Social Web". In: *The Semanic Web: Research and Applications: 8th Extended Semantic Web Conference, ESWC 2011*. Heraklion, Crete, Greece: Springer, pp. 375–389. ISBN: 3642210635.

Abel, Fabian, Claudia Hauff, Geert-Jan Houben, and Ke Tao (2012). "Leveraging user modeling on the Social Web with Linked Data". English. In: *Web Engineering: 12th International Conference, ICWE 2012*. Berlin, Germany: Springer, pp. 378–385.

Abel, Fabian, Eelco Herder, Geert-Jan Houben, Nicola Henze, and Daniel Krause (2013a). "Cross-system user modeling and personalization on the social web". In: *User Modeling and User-Adapted Interaction* 23.2-3, pp. 169–209. ISSN: 0924-1868.

Abel, Fabian, Qi Gao, Geert-Jan Houben, and Ke Tao (2013b). "Twitter-based User Modeling for News Recommendations". In: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. IJCAI '13. Beijing, China: AAAI Press, pp. 2962–2966. ISBN: 978-1-57735-633-2.

Abele, Andrejs, John P McCrae, Paul Buitelaar, Anja Jentzsch, and Richard Cyganiak (2017). "Linking Open Data cloud diagram (2017)". In: *2017-03-07]. http://lod-cloud. net*.

Ahmed, Amr, Yucheng Low, Mohamed Aly, Vanja Josifovski, and Alexander J Smola (2011). "Scalable distributed inference of dynamic user interests for behavioral targeting". In: *Proceedings of the 17th International Conference on Knowledge Discovery and Data Mining*. San Diego, California, USA: ACM, pp. 114–122. ISBN: 1450308139.

Ahn, Dabi, Taehun Kim, Soon J Hyun, and Dongman Lee (2012). "Inferring User Interest Using Familiarity and Topic Similarity with Social Neighbors in Facebook". In: *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01*. WI-IAT '12. Washington, DC, USA: IEEE Computer Society, pp. 196–200.

Alfarhood, S, K Labille, and S Gauch (2017a). "PLDSD: Propagated Linked Data Semantic Distance". In: *2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pp. 278–283. ISBN: VO -. DOI: 10.1109/WETICE.2017.16.

Alfarhood, Sultan, Susan Gauch, and Kevin Labille (2017b). "Employing Link Differentiation in Linked Data Semantic Distance". In: *Knowledge Engineering and Semantic Web*. Ed. by Przemysław Różewski and Christoph Lange. Cham: Springer International Publishing, pp. 175–191. ISBN: 978-3-319-69548-8.

Auer, Sören, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives (2007). "DBpedia: A nucleus for a web of open data". In: *The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference*. Busan, Korea: Springer, pp. 722–735. ISBN: 3540762973.

Berkovsky, Shlomo, Tsvi Kuflik, and Francesco Ricci (2008). "Mediation of user models for enhanced personalization in recommender systems". In: *User Modeling and User-Adapted Interaction* 18.3, pp. 245–286. ISSN: 0924-1868.

Berkovsky, Shlomo, Tsvi Kuflik, and Francesco Ricci (2009). "Cross-representation mediation of user models". In: *User Modeling and User-Adapted Interaction* 19.1, pp. 35–63. ISSN: 1573-1391. DOI: 10.1007/s11257-008-9055-z. URL: https://doi.org/10.1007/s11257-008-9055-z.

Besel, Christoph, Jörg Schlötterer, and Michael Granitzer (2016a). "Inferring semantic interest profiles from Twitter followees: Does Twitter know better than your friends?" In: *Proceedings of the 31st Annual ACM Symposium on Applied Computing*. SAC '16. New York, NY, USA: ACM, pp. 1152–1157. ISBN: 978-1-4503-3739-7.

Besel, Christoph, Jörg Schlötterer, and Michael Granitzer (2016b). "On the quality of semantic interest profiles for onine social network consumers". In: *ACM SIGAPP Applied Computing Review* 16.3, pp. 5–14. ISSN: 1559-6915.

Bhargava, Preeti, Oliver Brdiczka, and Michael Roberts (2015). "Unsupervised Modeling of Users' Interests from Their Facebook Profiles and Activities". In: *Proceedings of the 20th International Conference on Intelligent User Interfaces*. IUI '15. New York, NY, USA: ACM, pp. 191–201. ISBN: 978-1-4503-3306-1. DOI: 10.1145/2678025.2701365. URL: http://doi.acm.org/10.1145/2678025.2701365.

Bhattacharya, Parantapa, Muhammad Bilal Zafar, Niloy Ganguly, Saptarshi Ghosh, and Krishna P Gummadi (2014). "Inferring user interests in the Twitter social network". In: *Proceedings of the 8th ACM Conference on Recommender Systems*. RecSys'14. New York, NY, USA: ACM, pp. 357–360. ISBN: 978-1-4503-2668-1.

Bizer, Christian, Tom Heath, and Tim Berners-Lee (2009). "Linked Data - The Story So Far". In: *International Journal on Semantic Web and Information Systems* 5.3, pp. 1–22. DOI: 10.4018/jswis.2009081901.

Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). "Latent Dirichlet Allocation". In: *Journal of Machine Learning Research* 3.Jan, pp. 993–1022.

Bollacker, Kurt, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor (2008). "Freebase: a collaboratively created graph database for structuring human knowledge". In: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. AcM, pp. 1247–1250. ISBN: 160558102X.

Bordes, Antoine, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko (2013). "Translating Embeddings for Modeling Multi-relational Data". In: pp. 2787–2795.

Breslin, John G, Andreas Harth, Uldis Bojars, and Stefan Decker (2005). "Towards semantically-interlinked online communities". In: *The Semantic Web: Research and Applications*. Heraklion, Crete, Greece: Springer, pp. 500–514. ISBN: 3540261249.

Brickley, Dan and Libby Miller (2012). *FOAF vocabulary specification 0.98*. URL: http://xmlns.com/foaf/spec/ (visited on 09/22/2017).

Brusilovsky, Peter and Eva Millán (2007). "User models for adaptive hypermedia and adaptive educational systems". In: *The Adaptive Web*. Springer-Verlag, pp. 3–53. ISBN: 3540720782.

Brusilovsky, Peter, Alfred Kobsa, and Wolfgang Nejdl (2007). *The adaptive web: methods and strategies of web personalization*. Vol. 4321. Springer Science & Business Media. ISBN: 3540720782.

Budak, Ceren, Anitha Kannan, Rakesh Agrawal, and Jan Pedersen (2014). *Inferring user interests from microblogs*. Tech. rep. Microsoft.

Burel, Grégoire, Hassan Saif, and Harith Alani (2017). "Semantic Wide and Deep Learning for Detecting Crisis-Information Categories on Social Media BT - The Semantic Web – ISWC 2017: 16th International Semantic Web Conference, Vienna, Austria, October 21–25, 2017, Proceedings, Part I". In: ed. by Claudia D'Amato, Miriam Fernandez, Valentina Tamma, Freddy Lecue, Philippe Cudré-Mauroux, Juan Sequeda, Christoph Lange, and Jeff Heflin. Cham: Springer International Publishing, pp. 138–155. ISBN: 978-3-319-68288-4. DOI: 10.1007/978-3-319-68288-4_9.

Cantador, Iván, Martin Szomszor, Harith Alani, Miriam Fernández Sánchez, and Pablo Castells (2008). "Enriching ontological user profiles with tagging history for multi-domain recommendations". In: *CEUR Workshop Proceedings*. Yannis Avrithis. ISBN: 1613-0073.

Carmagnola, Francesca, Federica Cena, Luca Console, Omar Cortassa, Cristina Gena, Anna Goy, Ilaria Torre, Andrea Toso, and Fabiana Vernero (2008). "Tag-based user modeling for social multi-device adaptive guides". In: *User Modeling and User-Adapted Interaction* 18.5, pp. 497–538. ISSN: 1573-1391. DOI: 10.1007/s11257-008-9052-2. URL: http://link.springer.com/10.1007/s11257-008-9052-2.

Chen, Jilin, Rowan Nairn, Les Nelson, Michael Bernstein, and Ed Chi (2010). "Short and tweet: experiments on recommending content from information streams". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Atlanta, Georgia, USA: ACM, pp. 1185–1194. ISBN: 1605589292.

Cohen, Philip R and C Raymond Perrault (1979). "Elements of a plan-based theory of speech acts". In: *Cognitive Science* 3.3, pp. 177–212. ISSN: 0364-0213.

Collins, Allan M and Elizabeth F Loftus (1975). "A spreading-activation theory of semantic processing." In: *Psychological review* 82.6, p. 407. ISSN: 1939-1471.

Degemmis, Marco, Pasquale Lops, and Giovanni Semeraro (2007). "A content-collaborative recommender that exploits WordNet-based user profiles for neighborhood formation". In: *User Modeling and User-Adapted Interaction* 17.3, pp. 217–255. ISSN: 1573-1391. DOI: 10.1007/s11257-006-9023-4.

Di Noia, Tommaso, Roberto Mirizzi, Vito Claudio Ostuni, and Davide Romito (2012a). "Exploiting the Web of Data in Model-based Recommender Systems". In: *Proceedings of the 6th ACM Conference on Recommender Systems*. ACM, pp. 253–256. ISBN: 1450312705.

Di Noia, Tommaso, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito, and Markus Zanker (2012b). "Linked Open Data to Support Content-based Recommender Systems". In: *Proceedings of the 8th International*

Conference on Semantic Systems. ACM, pp. 1–8. ISBN: 978-1-4503-1112-0. DOI: 10.1145/2362499.2362501.

Di Noia, Tommaso, Iván Cantador, and Vito Claudio Ostuni (2014). "Linked Open Data-enabled Recommender Systems: ESWC 2014 Challenge on Book Recommendation". In: *Semantic Web Evaluation Challenge*. Springer, pp. 129–143. ISBN: 3319120239.

Dong, Anlei, Ruiqiang Zhang, Pranam Kolari, Jing Bai, Fernando Diaz, Yi Chang, Zhaohui Zheng, and Hongyuan Zha (2010). "Time is of the Essence: Improving Recency Ranking Using Twitter Data". In: *Proceedings of the 19th International Conference on World Wide Web*. WWW '10. New York, NY, USA: ACM, pp. 331–340. ISBN: 978-1-60558-799-8.

Drumond, Lucas, Steffen Rendle, and Lars Schmidt-Thieme (2012). "Predicting RDF triples in incomplete knowledge bases with tensor factorization". In: *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. ACM, pp. 326–331. ISBN: 1450308570.

Edmonds, Jack (1968). "Optimum branchings". In: *Mathematics and the Decision Sciences*, pp. 335–345.

Faralli, Stefano, Giovanni Stilo, and Paola Velardi (2015a). "Large scale homophily analysis in Twitter using a Twixonomy." In: *Proceedings of the 24th International Conference on Artificial Intelligence*. Buenos Aires, Argentina: AAAI Press, pp. 2334–2340.

Faralli, Stefano, Giovanni Stilo, and Paola Velardi (2015b). "Recommendation of Microblog Users based on Hierarchical Interest Profiles". In: *Social Network Analysis and Mining* 5.1, pp. 1–23. ISSN: 1869-5450.

Faralli, Stefano, Giovanni Stilo, and Paola Velardi (2017). "Automatic acquisition of a taxonomy of microblogs users' interests." In: *Web Semantics: Science, Services and Agents on the World Wide Web*. ISSN: 1570-8268. DOI: https://doi.org/10.1016/j.websem.2017.05.004.

Färber, Michael, Basil Ell, Carsten Menne, and Achim Rettinger (2015). "A comparative survey of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO". In: *Semantic Web Journal*, pp. 1–26.

Fernández, Javier D, Miguel A Martínez-Prieto, Claudio Gutiérrez, Axel Polleres, and Mario Arias (2013). "Binary RDF representation for publication and exchange (HDT)". In: *Web Semantics: Science, Services and Agents on the World Wide Web* 19, pp. 22–41. ISSN: 1570-8268. DOI: https://doi.org/10.1016/j.websem.2013.01.002. URL: http://www.sciencedirect.com/science/article/pii/S1570826813000036.

Figueroa, Cristhian, Iacopo Vagliano, Oscar Rodríguez Rocha, and Maurizio Morisio (2015). "A systematic literature review of Linked Data-based recommender systems". In: *Concurrency Computation*. ISSN: 15320634. DOI: 10.1002/cpe.3449. arXiv: arXiv:1302.5679v1.

Figueroa, Cristhian, Iacopo Vagliano, Oscar Rodriguez Rocha, Marco Torchiano, Catherine Faron Zucker, Juan Carlos Corrales, and Maurizio Morisio (2017). "Allied: A framework for executing linked data-based recommendation algorithms". In: *International Journal on Semantic Web and Information Systems (IJSWIS)* 13.4, pp. 134–154.

Flati, Tiziano, Daniele Vannella, Tommaso Pasini, and Roberto Navigli (2014). "Two is bigger (and better) than one: the Wikipedia bitaxonomy project." In: *52nd Annual Meeting of the Association for Computational Linguistics, ACL*. Baltimore, MD, United States: Association for Computational Linguistics (ACL), pp. 945–955.

Forbes, Peter and Mu Zhu (2011). "Content-boosted Matrix Factorization for Recommender Systems: Experiments with Recipe Recommendation". In: *Proceedings of the Fifth ACM Conference on Recommender Systems*. RecSys '11. New York, NY, USA: ACM, pp. 261–264. ISBN: 978-1-4503-0683-6. DOI: 10.1145/2043932.2043979. URL: http://doi.acm.org/10.1145/2043932.2043979.

Franz, Thomas, Antje Schultz, Sergej Sizov, and Steffen Staab (2009). "Triplerank: Ranking semantic web data by tensor decomposition". In: *The Semantic Web-ISWC 2009*, pp. 213–228.

Gabrilovich, Evgeniy and Shaul Markovitch (2007). "Computing semantic relatedness using wikipedia-based explicit semantic analysis." In: *Proceedings of the 20th International Joint Conference on Artifical Intelligence*. Hyderabad, India: Morgan Kaufmann Publishers Inc., pp. 1606–1611.

Galárraga, Luis, Simon Razniewski, Antoine Amarilli, and Fabian M Suchanek (2017). "Predicting Completeness in Knowledge Bases". In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, pp. 375–383. ISBN: 1450346758.

Gantner, Zeno, Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme (2011). "MyMediaLite: A Free Recommender System Library". In: *Proceedings of the Fifth ACM Conference on Recommender Systems*. RecSys '11. New York, NY, USA: ACM, pp. 305–308. ISBN: 978-1-4503-0683-6. DOI: 10.1145/2043932.2043989.

Gao, Qi, Fabian Abel, Geert-Jan Houben, and Ke Tao (2011). "Interweaving Trend and User Modeling for Personalized News Recommendation". In: *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01*. WI-IAT '11. Washington, DC, USA: IEEE Computer Society, pp. 100–103. ISBN: 978-0-7695-4513-4.

Gao, Qi, Fabian Abel, and Geert-Jan Houben (2012). "Genius: generic user modeling library for the social semantic web". In: *The semantic web*. Springer, pp. 160–175. ISBN: 3642299229.

Garcia Esparza, Sandra, Michael P O'Mahony, and Barry Smyth (2013). "CatStream: Categorising Tweets for User Profiling and Stream Filtering". In: *Proceedings of the 2013 International Conference on Intelligent User Interfaces*. IUI '13. New York, NY, USA: ACM, pp. 25–36. ISBN: 978-1-4503-1965-2.

Gauch, Susan, Mirco Speretta, Aravind Chandramouli, and Alessandro Micarelli (2007a). "The adaptive web". In: ed. by Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl. Berlin, Heidelberg: Springer-Verlag. Chap. User Profi, pp. 54–89. ISBN: 978-3-540-72078-2.

Gauch, Susan, Mirco Speretta, Aravind Chandramouli, and Alessandro Micarelli (2007b). "User profiles for personalized information access". In: *The adaptive web*. Springer, pp. 54–89. ISBN: 3540720782.

Gemmis, Marco de, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro (2015a). "Semantics-Aware Content-Based Recommender Systems". In: *Recommender Systems Handbook*. Springer, pp. 119–159. DOI: 10.1007/978-1-4899-7637-6_4.

Gemmis, Marco de, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro (2015b). "Semantics-Aware Content-Based Recommender Systems BT - Recommender Systems Handbook". In: ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Boston, MA: Springer US, pp. 119–159. ISBN: 978-1-4899-7637-6. DOI: 10.1007/978-1-4899-7637-6_4.

Gong, Wei, Ee-Peng Lim, and Feida Zhu (2015). "Characterizing Silent Users in Social Media Communities." In:

Goossen, Frank, Wouter IJntema, Flavius Frasincar, Frederik Hogenboom, and Uzay Kaymak (2011). "News personalization using the CF-IDF semantic recommender". In: *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*. ACM, p. 10. ISBN: 1450301487.

Große-Bölting, G, C Nishioka, and A Scherp (2015). *Generic process for extracting user profiles from social media using hierarchical knowledge bases*. Anaheim, CA, USA. DOI: 10.1109/ICOSC.2015.7050806.

Groues, Valentin, Yannick Naudet, and Odej Kao (2012). "Adaptation and evaluation of a semantic similarity measure for dbpedia: A first experiment". In: *Semantic and Social Media Adaptation and Personalization (SMAP)*. IEEE, pp. 87–91.

Guo, Shu, Quan Wang, Bin Wang, Lihong Wang, and Li Guo (2015). "Semantically Smooth Knowledge Graph Embedding." In: *ACL (1)*, pp. 84–94.

Haewoon, Kwak, Lee Changhyun, Park Hosung, and Moon Sue (2010). "What is Twitter, a social network or a news media?" In: *Proceedings of the 19th International Conference on World Wide Web*. Raleigh, North Carolina, USA: ACM.

Han, Lushan, Abhay L Kashyap, Tim Finin, James Mayfield, and Jonathan Weese (2013). "UMBC_EBIQUITY-CORE: Semantic Textual Similarity Systems." In: *The Second Joint Conference on Lexical and Computational Semantics*. Atlanta, GA, USA: Association for Computational Linguistics, pp. 44–52.

Hannon, John, Kevin McCarthy, Michael P O'Mahony, and Barry Smyth (2012). "A multi-faceted user model for twitter". In: *User Modeling, Adaptation, and Personalization: 20th International Conference, UMAP 2012*, Montreal, Canada: Springer, pp. 303–309. ISBN: 3642314538.

Haveliwala, Taher H (2003). "Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search". In: *IEEE transactions on knowledge and data engineering* 15.4, pp. 784–796. ISSN: 1041-4347.

Heath, Tom and Christian Bizer (2011). "Linked data: Evolving the web into a global data space". In: *Synthesis lectures on the semantic web: theory and technology* 1.1, pp. 1–136.

Heitmann, Benjamin (2012). "An open framework for multi-source, cross-domain personalisation with semantic interest graphs". In: *Proceedings of the sixth ACM conference on Recommender systems*. ACM, pp. 313–316. ISBN: 1450312705.

Heitmann, Benjamin and Conor Hayes (2010). "Using Linked Data to Build Open, Collaborative Recommender Systems". In: *AAAI spring symposium: linked data meets artificial intelligence*, pp. 76–81.

Heitmann, Benjamin and Conor Hayes (2014). "SemStim at the LOD-RecSys 2014 challenge". In: *Semantic Web Evaluation Challenge*. Springer, pp. 170–175. ISBN: 3319120239.

Hong, Liangjie, Aziz S Doumith, and Brian D Davison (2013). "Co-factorization machines: Modeling user interests and predicting individual decisions in Twitter". In: *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*. WSDM '13. New York, NY, USA: ACM, pp. 557–566. ISBN: 978-1-4503-1869-3.

Hung, Chia-Chuan, Yi-Ching Huang, Jane Yung-jen Hsu, and David Kuan-Chun Wu (2008). "Tag-based user profiling for social media recommendation". In: *Proceedings of the 21st ACM conference on Hypertext and hypermedia*, pp. 151–156.

Ingwersen, Peter (1994). "Polyrepresentation of information needs and semantic entities elements of a cognitive theory for information retrieval interaction". In: *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Dublin, Ireland: Springer, pp. 101–110.

Jain, Paridhi, Ponnurangam Kumaraguru, and Anupam Joshi (2013). "@i seek 'fb.me': identifying users across multiple online social networks". In:

Proceedings of the 22nd international conference on World Wide Web companion. ACM, pp. 1259–1268.

Jeh, Glen and Jennifer Widom (2002). "SimRank: A Measure of Structural-context Similarity". In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '02. New York, NY, USA: ACM, pp. 538–543. ISBN: 1-58113-567-X. DOI: 10.1145/775047.775126.

Ji, Guoliang, Kang Liu, Shizhu He, and Jun Zhao (2016). "Knowledge graph completion with adaptive sparse transfer matrix". In: *Thirtieth AAAI Conference on Artificial Intelligence*.

Jiang, Bo and Ying Sha (2015). "Modeling temporal dynamics of user interests in online social networks". In: *Procedia Computer Science* 51, pp. 503–512. ISSN: 1877-0509.

Kaminskas, Marius and Derek Bridge (2016). "Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems". In: *ACM Trans. Interact. Intell. Syst.* 7.1, 2:1–2:42. ISSN: 2160-6455. DOI: 10.1145/2926720. URL: http://doi.acm.org/10.1145/2926720.

Kang, Jaeyong and Hyunju Lee (2016). "Modeling User Interest in Social Media using News Media and Wikipedia". In: *Information Systems* 65, pp. 52–64. ISSN: 0306-4379.

Kanta, M, M Simko, and M Bieliková (2012). *Trend-Aware User Modeling with Location-Aware Trends on Twitter*. Luxembourg.

Kapanipathi, Pavan, Fabrizio Orlandi, Amit Sheth, and Alexandre Passant (2011). "Personalized Filtering of the Twitter Stream". In: *Proceedings of the Second International Conference on Semantic Personalized Information Management: Retrieval and Recommendation-Volume 781*. Bonn, Germany: CEUR-WS. org, pp. 6–13.

Kapanipathi, Pavan, Prateek Jain, Chitra Venkataramani, and Amit Sheth (2014). "User Interests Identification on Twitter Using a Hierarchical Knowledge Base". In: *The Semantic Web: Trends and Challenges*. Anissaras, Crete, Greece: Springer. Chap. 8, pp. 99–113.

Karatay, Deniz and Pinar Karagoz (2015). "User Interest Modeling in Twitter with Named Entity Recognition". In: *Making Sense of Microposts (# Microposts2015)*. Florence, Italy, pp. 17–20.

Kim, Dongwoo, Yohan Jo, Il-Chul Moon, and Alice Oh (2010). "Analysis of Twitter lists as a potential source for discovering latent characteristics of users". In: *ACM CHI Workshop on Microblogging*. Atlanta, Georgia, USA: Citeseer, p. 4.

Koren, Yehuda (2009). "Collaborative Filtering with Temporal Dynamics". In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge*

Discovery and Data Mining. KDD '09. New York, NY, USA: ACM, pp. 447–456. ISBN: 978-1-60558-495-9. DOI: 10.1145/1557019.1557072.

Lalithsena, Sarasi, Pavan Kapanipathi, and Amit Sheth (2016). "Harnessing Relationships for Domain-specific Subgraph Extraction: A Recommendation Use Case". In: *IEEE International Conference on Big Data*. Washington D.C.

Leacock, Claudia and Martin Chodorow (1998). "Combining Local Context and WordNet Similarity for Word Sense Identification". In: *An Electronic Lexical Database*, pp. 265–283.

Leal, José Paulo, Vânia Rodrigues, and Ricardo Queirós (2012). "Computing semantic relatedness using dbpedia". In: ISSN: 393989740X.

Lehmann, Jens, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, and Sören Auer (2013). "Dbpedia-a Large-scale, Multilingual Knowledge Base Extracted from Wikipedia". In: *Semantic Web Journal*.

Liao, Yang, Masud Moshtaghi, Bo Han, Shanika Karunasekera, Ramamohanarao Kotagiri, Timothy Baldwin, Aaron Harwood, and Philippa Pattison (2012). "Mining micro-blogs: opportunities and challenges". In: *Computational Social Networks*. Springer, pp. 129–159.

Lim, Kwan Hui and Amitava Datta (2013). "Interest Classification of Twitter Users Using Wikipedia". In: *Proceedings of the 9th International Symposium on Open Collaboration*. WikiSym '13. Hong Kong, China: ACM, 22:1–22:2. ISBN: 978-1-4503-1852-5.

Liu, Jing, Fan Zhang, Xinying Song, Young-In Song, Chin-Yew Lin, and Hsiao-Wuen Hon (2013). "What's in a name?: an unsupervised approach to link users across communities". In: *Proceedings of the sixth ACM International Conference on Web Search and Data Mining*. Rome, Italy: ACM, pp. 495–504. ISBN: 145031869X.

Locke, Brian William (2009). "Named entity recognition: Adapting to microblogging". PhD thesis.

Lu, Chunliang, Wai Lam, and Yingxiao Zhang (2012). "Twitter User Modeling and Tweets Recommendation based on Wikipedia Concept Graph". In: *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*. Toronto, Ontario, Canada.

Lv, Yuanhua and ChengXiang Zhai (2011). "Lower-bounding term frequency normalization". In: *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, pp. 7–16. ISBN: 1450307175.

Maedche, Alexander and Valentin Zacharias (2002). "Clustering ontology-based metadata in the semantic web". In: *Principles of Data Mining and Knowledge Discovery*. Springer, pp. 348–360. ISBN: 3540440372.

Manrique, Rubén and Olga Mariño (2017). "How Does the Size of a Document Affect Linked Open Data User Modeling Strategies?" In: *Proceedings*

of the International Conference on Web Intelligence. WI '17. New York, NY, USA: ACM, pp. 1246–1252. ISBN: 978-1-4503-4951-2.

Meymandpour, Rouzbeh and Joseph G Davis (2016). "A semantic similarity measure for linked data: An information content-based approach". In: *Knowledge-Based Systems* 109, pp. 276–293. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2016.07.012.

Mezghani, Manel, Corinne Amel Zayani, Ikram Amous, and Faiez Gargouri (2012). "A User Profile Modelling Using Social Annotations: A Survey". In: *Proceedings of the 21st International Conference on World Wide Web*. WWW '12 Companion. New York, NY, USA: ACM, pp. 969–976. ISBN: 978-1-4503-1230-1.

Michelson, Matthew and Sofus A Macskassy (2010). "Discovering Users' Topics of Interest on Twitter: A First Look". In: *Proceedings of the 4th Workshop on Analytics for Noisy Unstructured Text Data*. Toronto, ON, Canada: ACM, pp. 73–80. ISBN: 1450303765.

Mihalcea, Rada and Paul Tarau (2004). "TextRank: Bringing Order into Texts". In: *Proceedings of EMNLP 2004*. Ed. by Dekang Lin and Dekai Wu. Barcelona, Spain: Association for Computational Linguistics, pp. 404–411.

Miller, George A (1995). "WordNet: a lexical database for English". In: *Communications of the ACM* 38.11, pp. 39–41. ISSN: 0001-0782.

Musto, Cataldo, Pierpaolo Basile, Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro (2014). "Linked Open Data-enabled Strategies for Top-N Recommendations". In: *CBRecSys*, p. 49.

Musto, Cataldo, Fedelucio Narducci, Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro (2016a). "ExpLOD: A Framework for Explaining Recommendations Based on the Linked Open Data Cloud". In: *Proceedings of the 10th ACM Conference on Recommender Systems*. RecSys '16. New York, NY, USA: ACM, pp. 151–154. ISBN: 978-1-4503-4035-9. DOI: 10.1145/2959100.2959173.

Musto, Cataldo, Pasquale Lops, Pierpaolo Basile, Marco de Gemmis, and Giovanni Semeraro (2016b). "Semantics-aware Graph-based Recommender Systems Exploiting Linked Open Data". In: *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*. ACM, pp. 229–237. ISBN: 978-1-4503-4368-8. DOI: 10.1145/2930238.2930249.

Musto, Cataldo, Giovanni Semeraro, Marco de Gemmis, and Pasquale Lops (2017). "Tuning Personalized PageRank for Semantics-Aware Recommendations Based on Linked Open Data". In: *European Semantic Web Conference*. Cham: Springer, pp. 169–183. ISBN: 978-3-319-58068-5. DOI: 10.1007/978-3-319-58068-5_11.

Myers, Seth A and Jure Leskovec (2014). "The Bursty Dynamics of the Twitter Information Network". In: *Proceedings of the 23rd International Conference on World Wide Web*. Seoul, Korea: ACM, pp. 913–924. ISBN: 1450327443.

Narducci, Fedelucio, Cataldo Musto, Giovanni Semeraro, Pasquale Lops, and Marco Gemmis (2013). "Leveraging Encyclopedic Knowledge for Transparent and Serendipitous User Profiles". In: *User Modeling, Adaptation, and Personalization: 21th International Conference*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 350–352. ISBN: 6.

Nguyen, Phuong T, Paolo Tomeo, Tommaso Di Noia, and Eugenio Di Sciascio (2015). "Content-based recommendations via DBpedia and Freebase: a case study in the music domain". In: *International Semantic Web Conference*, pp. 605–621.

Nickel, Maximilian, Volker Tresp, and Hans-Peter Kriegel (2012). "Factorizing yago: scalable machine learning for linked data". In: *Proceedings of the 21st international conference on World Wide Web*. ACM, pp. 271–280. ISBN: 1450312292.

Nickel, Maximilian, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich (2016). "A Review of Relational Machine Learning for Knowledge Graphs". In: *Proceedings of the IEEE* 104.1, pp. 11–33. ISSN: 0018-9219.

Nishioka, Chifumi and Ansgar Scherp (2016). "Profiling vs. Time vs. Content: What Does Matter for Top-k Publication Recommendation Based on Twitter Profiles?" In: *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries*. JCDL '16. New York, NY, USA: ACM, pp. 171–180.

Nishioka, Chifumi, Gregor Große-Bölting, and Ansgar Scherp (2015). "Influence of Time on User Profiling and Recommending Researchers in Social Media". In: *Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business*. i-KNOW '15. New York, NY, USA: ACM, 9:1–9:8. ISBN: 978-1-4503-3721-2.

Noia, Tommaso Di, Vito Claudio Ostuni, Paolo Tomeo, and Eugenio Di Sciascio (2016). "Sprank: Semantic path-based ranking for top-n recommendations using linked open data". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 8.1, p. 9. ISSN: 2157-6904.

O'Banion, Shawn, Larry Birnbaum, and Kristian Hammond (2012). "Social media-driven news personalization". In: *Proceedings of the 4th ACM RecSys Workshop on Recommender Systems and the Social Web*. Dublin, Ireland: ACM, pp. 45–52. ISBN: 1450316387.

Obar, Jonathan A and Steven S Wildman (2015). "Social media definition and the governance challenge: An introduction to the special issue". In:

Oliveira, Jonice, Carla Delgado, and Ana Carolina Assaife (2017). "A Recommendation Approach for Consuming Linked Open Data". In: *Expert Systems with Applications* 72, pp. 407–420. ISSN: 0957-4174. DOI: http://dx.doi.org/10.1016/j.eswa.2016.10.037.

Orlandi, Fabrizio (2014). *Profiling user interests on the social semantic web*.

Orlandi, Fabrizio, John Breslin, and Alexandre Passant (2012). "Aggregated, Interoperable and Multi-domain User Profiles for the Social Web". In: *Proceedings of the 8th International Conference on Semantic Systems*. Graz, Austria: ACM, pp. 41–48.

Ostuni, Vito Claudio, Tommaso Di Noia, Eugenio Di Sciascio, and Roberto Mirizzi (2013). "Top-n Recommendations from Implicit Feedback Leveraging Linked Open Data". In: *Proceedings of the 7th ACM Conference on Recommender Systems*. ACM, pp. 85–92. ISBN: 1450324096.

Ostuni, Vito Claudio, Tommaso Di Noia, Roberto Mirizzi, and Eugenio Di Sciascio (2014). "A linked data recommender system using a neighborhood-based graph kernel". In: *E-Commerce and Web Technologies*. Springer, pp. 89–100. ISBN: 331910490X.

Page, Lawrence, Sergey Brin, Rajeev Motwani, and Terry Winograd (1999). *The PageRank citation ranking: Bringing order to the web.* Tech. rep.

Pan, Sinno Jialin and Qiang Yang (2010). "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10, pp. 1345–1359. ISSN: 1041-4347.

Passant, Alexandre (2010a). "dbrec: Music Recommendations Using DBpedia". English. In: *ISWC 2010 SE - 14*. Springer, pp. 209–224. ISBN: 978-3-642-17748-4. DOI: 10.1007/978-3-642-17749-1_14.

Passant, Alexandre (2010b). "Measuring semantic distance on linking data and using it for resources recommendations". In: *Proceedings of the AAAI Spring Symposium: Linked Data Meets Artificial Intelligence*. Vol. 77, pp. 93–98. ISBN: 9781577354611. URL: files/129/display.html.

Peñas, P, R del Hoyo, J Vea-Murguía, C González, and S Mayo (2013). *Collective Knowledge Ontology User Profiling for Twitter – Automatic User Profiling*. Atlanta, GA, USA.

Perera, Sujan, Pablo N Mendes, Adarsh Alex, Amit P Sheth, and Krishnaprasad Thirunarayan (2016). "Implicit Entity Linking in Tweets BT - The Semantic Web". In: *Latest Advances and New Domains: 13th International Conference, ESWC 2016*. Ed. by Harald Sack, Eva Blomqvist, Mathieu D'Aquin, Chiara Ghidini, Simone Paolo Ponzetto, and Christoph Lange. Cham: Springer International Publishing, pp. 118–132. ISBN: 978-3-319-34129-3.

Perrault, C Raymond, James F Allen, and Philip R Cohen (1978). "Speech acts as a basis for understanding dialogue coherence". In: *Proceedings of the 1978 Workshop on Theoretical issues in Natural Language Processing*. Association for Computational Linguistics, pp. 125–132.

Peska, Ladislav and Peter Vojtas (2013). "Using linked open data to improve recommending on e-commerce". In: *2nd International Workshop on Semantic Technologies meet Recommender Systems and Big Data*. CEUR-WS.org.

Phelan, Owen, Kevin McCarthy, and Barry Smyth (2009). "Using Twitter to Recommend Real-time Topical News". In: *Proceedings of the Third ACM Conference on Recommender Systems*. RecSys '09. New York, NY, USA: ACM, pp. 385–388. ISBN: 978-1-60558-435-5.

Piao, G. and J.G. Breslin (2018a). *Learning to rank tweets with author-based long short-term memory networks*. Vol. 10845 LNCS. ISBN: 9783319916613. DOI: 10.1007/978-3-319-91662-0_22.

Piao, Guangyuan (2016a). "Student Research Abstract: Exploiting the Semantic Similarity of Interests in a Semantic Interest Graph for Social Recommendations". In: *Proceedings of the 31st Annual ACM Symposium on Applied Computing*. SAC '16. New York, NY, USA: ACM, pp. 375–376. ISBN: 978-1-4503-3739-7. DOI: 10.1145/2851613.2852007. URL: http://doi.acm.org/10.1145/2851613.2852007.

Piao, Guangyuan (2016b). "Towards Comprehensive User Modeling on the Social Web for Personalized Link Recommendations". In: *User Modeling, Adaptation, and Personalization*. UMAP '16. Halifax, Nova Scotia, Canada: ACM, pp. 333–336. ISBN: 978-1-4503-4368-8. DOI: 10.1145/2930238.2930367.

Piao, Guangyuan and J.G. John G. Breslin (2016a). "Analyzing Aggregated Semantics-enabled User Modeling on Google+ and Twitter for Personalized Link Recommendations". In: *UMAP 2016 - Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*. Halifax, Nova Scotia, Canada: ACM, pp. 105–109. ISBN: 9781450343701. DOI: 10.1145/2930238.2930278.

Piao, Guangyuan and J.G. John G Breslin (2016b). "Exploring dynamics and semantics of user interests for user modeling on Twitter for link recommendations". In: *Proceedings of the 12th International Conference on Semantic Systems*. Vol. 13-14-Sept. ACM. Leipzig, Germany, pp. 81–88. ISBN: 9781450347525. DOI: 10.1145/2993318.2993332.

Piao, Guangyuan and J.G. John G Breslin (2016c). *Interest Representation, Enrichment, Dynamics, and Propagation: A Study of the Synergetic Effect of Different User Modeling Dimensions for Personalized Recommendations on Twitter*. Vol. 10024 LNAI. Bologna, Italy: Springer, pp. 496–510. ISBN: 9783319490038. DOI: 10.1007/978-3-319-49004-5_32.

Piao, Guangyuan and J.G. John G. Breslin (2016d). "User modeling on twitter with wordnet synsets and dbpedia concepts for personalized recommendations". In: *International Conference on Information and Knowledge Management, Proceedings*. Vol. 24-28-Octo. Indianapolis, Indiana, USA: ACM, pp. 2057–2060. ISBN: 9781450340731. DOI: 10.1145/2983323.2983908.

Piao, Guangyuan and J.G. John G. Breslin (2017a). *Factorization machines leveraging lightweight linked open data-enabled features for top-N recommendations*.

Vol. 10570 LNCS. Springer. ISBN: 9783319687858. DOI: 10.1007/978-3-319-68786-5_33.

Piao, Guangyuan and J.G. John G. Breslin (2017b). *Inferring User Interests for Passive Users on Twitter by Leveraging Followee Biographies*. Vol. 10193 LNCS. Aberdeen, UK: Springer. ISBN: 9783319566078. DOI: 10.1007/978-3-319-56608-5_10.

Piao, Guangyuan and J.G. John G. Breslin (2017c). "Leveraging Followee List Memberships for Inferring User Interests for Passive Users on Twitter". In: *HT 2017 - Proceedings of the 28th ACM Conference on Hypertext and Social Media*. Prague, Czech Republic: ACM Press. ISBN: 9781450347082. DOI: 10.1145/3078714.3078730.

Piao, Guangyuan and John G. Breslin (2015). "DBQuote: A Social Web based System for Collecting and Sharing Wisdom Quotes". In: *Proceedings of the 5th Joint International Semantic Technology Conference, Poster and Demonstrations*.

Piao, Guangyuan and John G. Breslin (2016e). "Analyzing MOOC Entries of Professionals on LinkedIn for User Modeling and Personalized MOOC Recommendations". In: *24th Conference on User Modeling, Adaptation and Personalization*. ACM.

Piao, Guangyuan and John G. Breslin (2016f). "Measuring semantic distance for linked open data-enabled recommender systems". In: *Proceedings of the 31st Annual ACM Symposium on Applied Computing*. Vol. 04-08-Apri. Pisa, Italy: ACM, pp. 315–320. ISBN: 9781450337397. DOI: 10.1145/2851613.2851839.

Piao, Guangyuan and John G. Breslin (2018b). "A Study of the Similarities of Entity Embeddings Learned from Different Aspects of a Knowledge Base for Item Recommendations". In: *The 1st Workshop on Deep Learning for Knowledge Graphs and Semantic Technologies at the 15th Extended Semantic Web Conference*.

Piao, Guangyuan and John G. Breslin (2018c). "Domain-Aware Sentiment Classification using GRUs and CNNs". In: *1st Place in the Semantic Sentiment Analysis Challenge at the 15th Extended Semantic Web Conference*. Springer.

Piao, Guangyuan and John G. Breslin (2018d). "Financial Aspect and Sentiment Predictions with Deep Neural Networks: An Ensemble Approach". In: *Financial Opinion Mining and Question Answering Workshop at The Web Conference (WWW)*. ACM.

Piao, Guangyuan and John G Breslin (2018e). "Inferring User Interests in Microblogging Social Networks: A Survey (accepted)". In: *User Modeling and User-Adapted Interaction*. arXiv: 1712.07691. URL: http://arxiv.org/abs/1712.07691.

Piao, Guangyuan and John G. Breslin (2018f). "Transfer Learning for Item Recommendations and Knowledge Graph Completion in Item Related Domains via a Co-Factorization Model". In: *The 15th Extended Semantic Web Conference*. Springer.

Piao, Guangyuan, Safina showkat Ara, and John G. Breslin (2015). "Computing the Semantic Similarity of Resources in DBpedia for Recommendation Purposes". In: *Semantic Technology*. Springer International Publishing, pp. 1–16. DOI: 10.1007/978-3-319-31676-5_13.

Qu, Zhonghua and Yang Liu (2011). "Interactive group suggesting for Twitter". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics, pp. 519–523. ISBN: 1932432884.

Rendle, Steffen (2010). "Factorization machines". In: *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, pp. 995–1000. ISBN: 1424491312.

Rendle, Steffen (2012). "Factorization Machines with libFM". In: *ACM Trans. Intell. Syst. Technol.* 3.3, 57:1–57:22. ISSN: 2157-6904. DOI: 10.1145/2168752.2168771.

Rendle, Steffen and Lars Schmidt-Thieme (2010). "Pairwise Interaction Tensor Factorization for Personalized Tag Recommendation". In: *Proceedings of the Third ACM International Conference on Web Search and Data Mining*. ACM, pp. 81–90. ISBN: 978-1-60558-889-6. DOI: 10.1145/1718487.1718498.

Rendle, Steffen, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme (2009). "BPR: Bayesian Personalized Ranking from Implicit Feedback". In: *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. AUAI Press, pp. 452–461. ISBN: 978-0-9749039-5-8.

Ricci, Francesco, Lior Rokach, and Bracha Shapira (2011). *Introduction to recommender systems handbook*. Springer. ISBN: 0387858199.

Rich, Elaine (1979). "User modeling via stereotypes*". In: *Cognitive Science* 3.4, pp. 329–354. ISSN: 1551-6709.

Rosen-Zvi, Michal, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth (2004). "The Author-topic Model for Authors and Documents". In: *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*. UAI '04. Arlington, Virginia, United States: AUAI Press, pp. 487–494. ISBN: 0-9749039-0-6.

Rotta, Randolf and Andreas Noack (2011). "Multilevel Local Search Algorithms for Modularity Clustering". In: *Journal of Experimental Algorithmics* 16, 2.3:2.1–2.3:2.27. ISSN: 1084-6654. DOI: 10.1145/1963190.1970376.

Rowe, Matthew (2014). "Transferring Semantic Categories with Vertex Kernels: Recommendations with SemanticSVD++". English. In: *The Semantic Web – ISWC 2014: 13th International Semantic Web Conference*. Ed. by Peter

Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig Knoblock, Denny Vrandečić, Paul Groth, Natasha Noy, Krzysztof Janowicz, and Carole Goble. Vol. 8796. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 341–356. ISBN: 978-3-319-11964-9. DOI: 10.1007/978-3-319-11964-9_22. URL: http://dx.doi.org/10.1007/978-3-319-11964-9{_}22.

Sakaki, Takeshi, Makoto Okazaki, and Yutaka Matsuo (2010). "Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors". In: *Proceedings of the 19th International Conference on World Wide Web*. WWW '10. New York, NY, USA: ACM, pp. 851–860. ISBN: 978-1-60558-799-8.

Salton, Gerard and Michael J McGill (1986). *Introduction to Modern information Retrieval*. McGraw-Hill, Inc.

Sang, Jitao, Dongyuan Lu, and Changsheng Xu (2015). "A Probabilistic Framework for Temporal User Modeling on Microblogs". In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. CIKM '15. New York, NY, USA: ACM, pp. 961–970. ISBN: 978-1-4503-3794-6. DOI: 10.1145/2806416.2806470. URL: http://doi.acm.org/10.1145/2806416.2806470.

Schein, Andrew I, Alexandrin Popescul, Lyle H Ungar, and David M Pennock (2002). "Methods and metrics for cold-start recommendations". In: *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 253–260. ISBN: 1581135610.

Sheth, Amit and Pavan Kapanipathi (2016). "Semantic Filtering for Social Data". In: *IEEE Internet Computing* 20.4, pp. 74–78. ISSN: 1089-7801.

Siehndel, Patrick and Ricardo Kawase (2012). "TwikiMe!: user profiles that make sense". In: *Proceedings of the 2012th International Conference on Semantic Web (Posters and Demonstrations Track) - Volume 914*. ISWC-PD'12. CEUR-WS.org, pp. 61–64.

Stefani, Anna (1998). "Personalizing access to web sites: The SiteIF project". In: *Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia HYPERTEXT*.

Strobin, Lukasz and Adam Niewiadomski (2013). "Evaluating semantic similarity with a new method of path analysis in RDF using genetic algorithms". In: *COMPUTER SCIENCE* 21.2, pp. 137–152.

Suchanek, Fabian M, Gjergji Kasneci, and Gerhard Weikum (2007). "Yago: a core of semantic knowledge". In: *Proceedings of the 16th international conference on World Wide Web*. ACM, pp. 697–706. ISBN: 1595936548.

Szomszor, Martin, Harith Alani, Ivan Cantador, Kieron O'Hara, and Nigel Shadbolt (2008). "Semantic modelling of user interests based on cross-folksonomy analysis". English. In: *The Semantic Web - ISWC 2008 SE - 40*.

Vol. 5318. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 632–648. ISBN: 3540885633.

Tao, Ke, Fabian Abel, Qi Gao, and Geert-Jan Houben (2012). "TUMS: Twitter-Based User Modeling Service". In: *The Semantic Web: ESWC 2011 Workshops*. Ed. by Raúl García-Castro, Dieter Fensel, and Grigoris Antoniou. Vol. 7117. Springer Berlin Heidelberg. Chap. 22, pp. 269–283.

Thalhammer, Andreas and Achim Rettinger (2016). "PageRank on Wikipedia: Towards General Importance Scores for Entities". In: *The Semantic Web: ESWC 2016 Satellite Events, Revised Selected Papers*. Cham: Springer International Publishing, pp. 227–240. ISBN: 978-3-319-47602-5. DOI: 10.1007/978-3-319-47602-5_41.

Tommaso, Giorgia Di, Stefano Faralli, Giovanni Stilo, and Paola Velardi (2018). "WIKI-MID: A VERY LARGE MULTI-DOMAIN INTERESTS DATASET OF TWITTER USERS WITH MAPPINGS TO WIKIPEDIA". In: *The 17th International Semantic Web Conference*. Springer.

Vrandečić, Denny and Markus Krötzsch (2014). "Wikidata: a Free Collaborative Knowledgebase". In: *Communications of the ACM* 57.10, pp. 78–85. ISSN: 0001-0782.

Vu, Thuy and Victor Perez (2013). "Interest Mining from User Tweets". In: *Proceedings of the 22Nd ACM International Conference on Information and Knowledge Management*. CIKM '13. New York, NY, USA: ACM, pp. 1869–1872. ISBN: 978-1-4503-2263-8.

Wang, Zhen, Jianwen Zhang, Jianlin Feng, and Zheng Chen (2014). "Knowledge Graph Embedding by Translating on Hyperplanes." In: *AAAI*, pp. 1112–1119.

Weng, Jianshu, Ee-Peng Lim, Jing Jiang, and Qi He (2010). "TwitterRank: Finding Topic-sensitive Influential Twitterers". In: *Proceedings of the Third ACM International Conference on Web Search and Data Mining*. WSDM '10. New York, NY, USA: ACM, pp. 261–270. ISBN: 978-1-60558-889-6.

White, Ryen W, Peter Bailey, and Liwei Chen (2009). "Predicting user interests from contextual information". In: *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '09. New York, NY, USA: ACM, pp. 363–370. ISBN: 978-1-60558-483-6.

Wu, Qiang, Christopher J C Burges, Krysta M Svore, and Jianfeng Gao (2010). "Adapting Boosting for Information Retrieval Measures". In: *Information Retrieval* 13.3, pp. 254–270. ISSN: 1386-4564.

Xu, Zhiheng, Long Ru, Liang Xiang, and Qing Yang (2011). "Discovering user interest on twitter with a modified author-topic model". In: *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*. Washington, DC, USA: IEEE Computer Society, pp. 422–429. ISBN: 0769545130.

Zarrinkalam, Fattane (2015). "Semantics-Enabled User Interest Mining". English. In: *The Semantic Web. Latest Advances and New Domains SE - 54*. Ed. by Fabien Gandon, Marta Sabou, Harald Sack, Claudia D'Amato, Philippe Cudré-Mauroux, and Antoine Zimmermann. Vol. 9088. Lecture Notes in Computer Science. Springer International Publishing, pp. 817–828. ISBN: 978-3-319-18817-1.

Zarrinkalam, Fattane and Mohsen Kahani (2015). "Semantics-enabled User Interest Detection from Twitter". In: *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. Singapore, pp. 469–476.

Zarrinkalam, Fattane, Hossein Fani, Ebrahim Bagheri, and Mohsen Kahani (2016). "Inferring Implicit Topical Interests on Twitter". In: *European Conference on Information Retrieval*. Padua, Italy: Springer, pp. 479–491.