UNIVERSITÀ DEGLI STUDI
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE
**ICT International Doctoral School**

# Linking Knowledge Bases to Social Media Profiles

Yaroslav Nechaev

*Advisor*

Dr. **Claudio Giuliano**
Fondazione Bruno Kessler

*Co-Advisor*

Dr. **Francesco Corcoglioniti**
Fondazione Bruno Kessler

*Committee*

Dr. **Serena Villata**
CNRS, France

Prof. Dr. **Simone Paolo Ponzetto**
Universität Mannheim, Germany

Prof. Dr. **Elena Cabrio**
Université Côte d'Azur, France

Prof. Dr. **Marco Rospocher**
Università degli studi di Verona

April 2019

# Abstract

The Linked Open Data (LOD) cloud is currently a primary source of background knowledge for tasks in a wide variety of domains and across many scientific fields. The structured nature and the usage of well-defined open standards make it convenient to contribute to and build upon. However, since the major part of the LOD is ultimately crowdsourced and mostly populated and updated manually, some of the content in the LOD can become stale, inconsistent and lack coverage. Social media, on the other hand, uniquely allow the real world events to be accurately reflected with little or no delay in the form of posts and profile updates. A major downside of this vibrant source of knowledge that is contained in the social media is its lack of structure, significant noisiness and restrictive APIs that make it hard to extract, analyze and use it in the downstream tasks.

In this thesis, I present the task of linking entities in a knowledge base (KB) to the corresponding social media profiles as an attempt to bridge the structured LOD cloud and the vibrant social media. As will be shown, such linking allows knowledge transfer between the two worlds: on the one hand, enabling the Semantic Web practitioners to harvest this vast amount of valuable, up-to-date data from the social media; on the other hand, the social media researchers can use the structured LOD knowledge much more efficiently, simplifying the pipelines and improving performance for tasks such as Type Prediction, Entity Linking, and User Profiling. I implement such knowledge transfer using DBpedia as a KB, since it is a cornerstone dataset in the LOD, and Twitter as a social media, due to its popularity and relative accessibility. However, approaches developed here are designed to be general and could be applied to other social media and KBs.

To this end, firstly, I introduce SocialLink — a project designed to link KBs to social media profiles. SocialLink consists of (i) a linking approach that is able to produce high-quality entity-profile pairs, (ii) a LOD-compliant dataset of alignments between DBpedia and Twitter, (iii) the Social Media Toolkit (SMT) system providing additional functionality on top of SocialLink. SocialLink employs a custom deep neural network-based architecture designed to efficiently exploit many modalities of data representing entities and profiles within DBpedia and Twitter.

In second, I demonstrate how SocialLink can facilitate tasks in both Semantic Web and Social Media Analysis. In particular, I employ the abovementioned knowledge transfer to achieve state-of-the-art performance in Type Prediction task on DBpedia. Additionally, SocialLink is used to infer user interests on Twitter and to implement a novel approach that I proposed to prevent such inference. Finally, the Entity Linking capabilities of SocialLink are exploited to augment the social media management application called Pokedem and to provide an additional performance boost to a conventional Entity Linking pipeline achieving the second-best performance in EVALITA 2016 competition.

SocialLink[1] and its applications are open source projects with all the code, datasets, tutorials and experimental results available online. The approaches presented in this thesis have been validated with extensive evaluation and covered in a number of publications.

**Keywords**:
Machine Learning, Social Media, Semantic Web, User Profiling, Entity Matching, Deep Learning

[1] http://sociallink.futuro.media/

# Acknowledgments

It wouldn't be possible for me to complete such an undertaking without the support of many people. Firstly, I would like to thank my advisors: Dr. Claudio Giuliano and Dr. Francesco Corcoglioniti. Their mentorship was instrumental, and I wouldn't have been able to come all this way without both. They've taught me how to conduct high-quality research; provided guidance and/or contributions to every paper I have written and every piece of software I've developed during my studies. Many of the skills that allowed me to become a scientist were cultivated by them.

I thank Prof. Elena Cabrio and Dr. Fabien Gandon for the short time that I've spent in Inria Sophia-Antipolis. Internship there allowed me to explore new research directions. I thank my colleagues at Fondazione Bruno Kessler and the University of Trento for fruitful collaborations, important discussions, and great feedback. Additionally, I'd like to recognize Prof. Alessandro Moschitti for his feedback on my initial research proposal.

I thank Prof. Igor Golovin, who was my advisor during my studies at Moscow State University, for believing in my abilities and for introducing me to the fields of Machine Learning and Natural Language Processing. Additionally, I thank Prof. Alexander Morozov for his mentorship during my internship at Rusnano.

I'd like to thank my family; their support was of paramount importance to me. I thank my wife Yulia for her love, for not letting me give up and reading and reviewing all of my papers. I'm grateful to my parents for their support that allowed me to move forward and pursue my interests to the best of my ability. I thank my grandfather, Dr. Evgeniy Nechaev, for setting an example of what being a professional means.

Finally, I want to thank my friends, especially Sergey Ivannikov and Dr. Dmitry Alexeev for their friendship and constant support, for inspiring me to pursue the doctoral degree in the first place and providing a healthy sense of rivalry during this period.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

This thesis is aimed to bring together Semantic Web and Social Media Analysis by enabling the knowledge transfer between the two and, as a result, improving approaches in tasks, such as type prediction, named entity linking and user profiling. In this chapter, first I introduce the relevant context (Section 1.1). Then, I describe the problems that are being addressed in this thesis (Section 1.2). Finally, I present the contributions (Section 1.3), structure (Section 1.4), list of publications that support the contributions (Section 1.5) and the artifacts (Section 1.6) of this thesis.

## 1.1 Context

Today it is hard to imagine a public person or an organization that does not have a social media account. Such entities typically have a rich presence in the social media, sharing content, engaging with their audience, maintaining and expanding their popularity. They post new content frequently and keep all the information in their profiles as relevant and precise as possible so that a potential consumer or a fan can be informed about the latest developments in no time. Thus, social media have become a primary source of information providing up-to-date knowledge on a wide variety of topics. An enormous amount of posts, profile updates, comments, images, and videos are being produced each day in response to real-world events ranging from the ones of planet-wide importance like the Olympics all the way to minor local or even personal events. Additionally, structured and semistructured data is voluntarily being filled by users: from the opening hours of stores to what books or songs a particular celebrity likes. Given that Facebook alone has more than 2B monthly active users, the scale and the coverage of such data are hard to overestimate. A desire to benefit from such vast array of knowledge have fueled a more than a decade-long research interest in social media in different scientific communities and companies. While extracting, storing, processing and analyzing the semi-structured data at such a scale is challenging, countless approaches were proposed over the years to

tackle various tasks in social media analysis, such as user profiling, profile matching, entity linking, community detection and labeling and many others.

On the other hand, from the very inception of the world wide web, there have been various efforts to gather and systematize human knowledge. One of the most prominent examples, Wikipedia, the largest crowdsourced online encyclopedia, presents such knowledge in a convenient, human-readable fashion. Complementary to this, the Semantic Web community has spawned the Linked Open Data (LOD) project to represent the same knowledge and much more in a machine-readable form. The LOD cloud is a collection of interlinked, standardized and structured datasets, or Knowledge Bases (KB), that have long become an invaluable source of data for many tasks, especially in data mining and knowledge discovery. A Knowledge Base contains a slice of humanity's knowledge in the form of a Knowledge Graph (KG), where entities participate in relationships according to some formal description called ontology and are encoded using RDF as a backbone. The well-understood process of processing the RDF data and the public availability of LOD datasets have made them staple in pipelines that can benefit from the use of large quantities of background knowledge. The community effort has created many billions of RDF triples coming from hundreds of resources over the years. Notoriously, Wikipedia-based knowledge bases, such as DBpedia, YAGO, and Wikidata, created from the labor of millions[1] of Wikipedia editors, are often used in a wide range of Natural Language Processing (NLP) tasks.

Data in social media and KBs present opposite characteristics. On the one hand, KBs provide high-quality structured information (e.g., YAGO has 95% accuracy) that is easily accessible, while data from social media is often noisy, unstructured, and hidden behind restrictive APIs. To extract from the social media as much information as contained in a typical KB entry sophisticated pipelines had to be built. As a result, significant research effort went towards solving tasks on social media, such as event detection, user profiling, and entity linking. These tasks mainly have to exploit supervised learning, which requires training sets that are scarcely available and expensive to create manually. On the other hand, social media provide up-to-date (real-time) information, while contents in KBs may lag behind from hours to months. For Wikipedia-related KBs, such lag comprises both the time for changing the page (hours to months, based on popularity) and, for automatically extracted KBs like DBpedia and YAGO, the time for that change to propagate in the KB (months to years). Such lag may prevent using these KBs in some application scenarios.

Coincidentally, for entities that exist in both the social media and in the KB (mainly people and organizations), the knowledge extracted from the one can complement and confirm the data in other. Data from the social media can update stale entries, fill the

[1] https://en.wikipedia.org/wiki/Wikipedia:Authors_of_Wikipedia

blanks in the KB and act as a reference supporting an existing fact in the KB,[2] while KBs can provide a solid structured background to the analysis performed on the social media. This idea that in essence is a knowledge transfer between the vibrant social media world and the structured Linked Open Data cloud forms the foundation of this thesis. Some studies have already exploited the LOD to augment social media analysis pipelines (Piao and Breslin 2017; Besel et al. 2016) providing the initial proof that such transfer is feasible. The aim of this thesis is to investigate the ways to enable this knowledge transfer and capitalize on it by improving and augmenting existing pipelines in both the Semantic Web and Social Media Analysis fields using the data from the social media and the background knowledge from the KBs respectively.

While the LOD cloud currently contains over 1,000 linked datasets[3] and there is a multitude of social media, in this thesis I specifically explore the knowledge transfer between DBpedia and Twitter. DBpedia is a general purpose KB derived from Wikipedia that forms a foundation of the LOD cloud. It is one of the largest and most popular datasets in the LOD and is the most interlinked one making it a perfect choice for the task at hand. Twitter is one of the major social media having more than 350M monthly active users (statista.com, 2019c). It has the least restrictive open API among big commercial social networks allowing easy access to a significant portion of data and is vastly popular among researchers across many areas of Computer and Social Sciences. Additionally, Twitter functionality is a bare minimum of what the social network typically is (includes the notions of posts, social graph, semi-structured profile, shares, likes and mentions), meaning that many of the approaches designed for Twitter would work on other social media with little or no modification. In sum, bridging DBpedia and Twitter provides the most potential research impact while minimizing the cost of acquiring the needed data and doing experiments.

## 1.2 Problem definition

In order to enable the knowledge transfer between KBs and social media, a significant amount of links between them has to be present effectively acting as a "bridge" connecting the two worlds. There are more than two million living people and more than half a million of currently existing organizations listed in DBpedia, many of which would have some presence in social media. Given that there are just 56,113 existing links from such entities in Wikidata and DBpedia to Twitter, one could expect a much more significant coverage

[2]Since an account in social media can be designated as official means of communication for an entity in the KB, information extracted from the content that is produced by this account can corroborate facts in the KB.

[3]1,229 datasets as of November 2018 (see https://lod-cloud.net)

to perform the knowledge transfer and to benefit from it successfully. As will be shown in Chapter 5, the amount of available links between KBs and social media significantly affects the performance of the downstream tasks that would like to benefit from such knowledge transfer. In order to overcome this disconnect, in this thesis, I present ways to establish additional links and, as a result, the potential link coverage between DBpedia and Twitter is increased more than tenfold.

With this in mind, I present the task of linking knowledge base entities to social media profiles which is at the core of this work. The goal is to find a profile in a social media for a given entity in a Knowledge Base. The task could be formulated the other way around, i.e., to find an entity in a KB for the chosen social media profile. However, given that there are billions of profiles in a major social media, which can only be partially acquired via expensive crawling, the inverse task will not be touched in this thesis.[4] As mentioned above, in this thesis, I mainly discuss such linking from DBpedia to Twitter, although the task defined here and most of the contributions are general and may apply to other KBs and social media with similar characteristics. To further limit the scope, I will only focus on entities for living people and currently existing organizations. While correct links to social media may be established for other entity types, such as brands, products and events, and some profiles of the deceased people and dissolved organizations are preserved in the social media, their linking will not be covered in this thesis. In DBpedia version 2016-04, this limits the task to a little more than 2.5M entities (see Chapter 4).

While the linking problem presented here is similar to the tasks of entity matching and profile matching that are well-studied by scholars in the fields of Semantic Web (among others) and Social Media Analysis respectively, unique challenges arising from aligning such vastly different resources warrant for a custom solution. Indeed, for entity matching — i.e., the task of finding KB entities referring to the same real-world entity – it is typically assumed to have *structured* information about *all* the entities to be matched in advance, which means that a social network would have to be fully available in some structured form to be amenable to entity matching techniques. As social media are neither openly available nor structured entity matching methods cannot be applied as is. On the other hand, research in profile matching, which is the task of aligning profiles in multiple social networks that correspond to a single person, can not be applied due to its reliance on behavioural patterns that people exhibit when creating multiple social media profiles (e.g., similar social graphs, similarities in a chosen account handle). Such patterns are not typically available in KBs.

Designing a linking approach also requires taking into consideration issues and peculiarities of processing the LOD and social media data. Firstly, different types of information

[4]Nevertheless, some recent works (Besel et al., 2016; Piao and Breslin, 2017) were able to successfully tackle this task for a small subset of Twitter users.

are available. Twitter mainly contains significant amounts of unstructured textual and media data, while DBpedia offers a wide range of structured properties that, unfortunately, can not be aligned to Twitter. Secondly, the quality and the amount of data is varying significantly from entity to entity and among different social profiles. DBpedia suffers from the knowledge lag discussed above, while Twitter data may be noisy, incomplete or deliberately false. In third, Twitter contains an enormous amount of accounts inevitably increasing ambiguity, which requires the approach to be more conservative to take into consideration possible fake and fan accounts, namesakes and even multiple true profiles. All of these issues and challenges will be covered in details in Chapter 3.

After enough links between a KB and a social network are populated, existing approaches for the variety of tasks in both can be modified to benefit from the knowledge transfer. Type prediction, which is the task of predicting missing type information for entities in the KB, is a prominent example. Indeed, as seen in Aprosio et al. (2013), in addition to the target KB's knowledge graph itself, type prediction approaches can exhibit improved performance by ingesting aligned data from other sources. Important tasks in social media can exploit the LOD to improve and simplify approaches as well. User profiling is the task of predicting a target user attribute given all the information that is known about the user. While being immensely popular among researchers in many fields, this task is essential for the commercial success of social media and the surrounding infrastructure. In Besel et al. (2016) and Piao and Breslin (2017), data imported from Wikipedia through Twitter-Wikipedia links is used as an essential piece of the profiling pipeline. In this thesis, I modify the approaches for both tasks to benefit from the data that can be imported along the Twitter-DBpedia links populated by the approach detailed in this thesis.

Moreover, the ingestion of data through the newly found links not only helps with the raw performance of the current systems, but more importantly, it enables new directions to be explored and novel approaches to be proposed that could not be made working by using the existing links. For example, Named Entity Linking (NEL), which is the task of disambiguating the identity of entities mentioned in the text, is typically designed to align entities in the text to entities in some KB. In this thesis, I demonstrate that the populated links and even the proposed linking approach itself can be used to complement typical NEL pipelines and enable linking to social media profiles instead of KB entries.

In sum, the current research efforts to populate and maintain the LOD cloud and the ever-increasing research interest in analyzing social media can both benefit from the knowledge transfer between those two worlds. However, such transfer can only function given that a significant amount of links between the KBs and the social networks can be established and the corresponding approaches are appropriately updated to exploit them.

## 1.3 Contributions

The core contribution of this thesis is SocialLink — the approach that is designed to align knowledge base entities to social media profiles and the public LOD dataset that contains precomputed alignments between DBpedia and Twitter. SocialLink effectively bridges the two worlds by (i) providing a scalable and robust machine learning-based procedure to find possible alignments for a given knowledge base entity and by (ii) providing a LOD compliant resource that can be used by the Semantic Web practitioners and the social media researchers alike without the need of instantiating the entire linking pipeline. Such bridge enables knowledge transfer in both directions and the approach itself can be used to link social media profiles to entities outside of the LOD cloud. SocialLink approach is a deep neural network-based system that is able to efficiently encode and utilize multiple modalities of data. For example, SocialLink exploits dense embeddings to represent knowledge and social graphs acquired independently in an unsupervised way, and it is then able to learn the similarity function to perform the alignment. Textual and nominal features are combined to assist in the task as well. As of version $v3.0$, released in October 2018, SocialLink performs linking starting from 2.5M entities found in 120 DBpedia language chapters and considers 291M Twitter users for alignment. This results in high quality links for 322K entities and lists candidate alignments with varying levels of confidence for over 600K more entities, providing more than tenfold increase in the number of DBpedia-Twitter links that were available in the LOD before.

SocialLink is an open source project that has been updated with algorithm and pipeline improvements and the dataset releases. The code is publicly available on Github,[5] while all the datasets are released on Figshare (Nechaev et al., 2017c) and Zenodo (Nechaev et al., 2018a). The website[6] contains all of the above along with additional resources, documentation and a public SPARQL endpoint to simplify and facilitate the usage of the dataset. SocialLink slowly but steadily gains adoption. Multiple teams in EVALITA2016 Entity Linking challenge were able to improve their results by exploiting the populated links. Additionally, Wikimedia Foundation has awarded the largest project grant of 2017 to **soweego**:[7] a project with the goal of incorporating SocialLink's linking pipeline as part of Wikidata to automatically align external catalogs and to provide references to claims contained in it.

In order to prove SocialLink's ability to provide added value via the aforementioned knowledge transfer, a number of additional contributions are presented in this thesis. Firstly, a *type prediction* approach was developed using social media data injected along

[5] https://github.com/Remper/sociallink
[6] https://w3id.org/sociallink
[7] https://meta.wikimedia.org/wiki/Grants:Project/Hjfocs/soweego

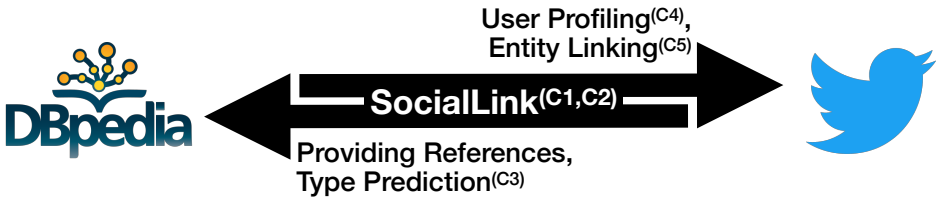



Figure 1.1. A diagram of thesis contributions: SocialLink enables knowledge transfer between the Linked Open Data cloud and social media facilitating tasks in both directions. Automatic referencing of claims in the KB is not a separate contribution of this thesis but it will be mentioned in Chapter 4 as part of the discussion around the **soweego** project.

the populated links. Typical approaches for predicting types in the LOD utilize information already present in the knowledge graph to infer the missing type relations. Alternatively, the information from the related resources, such as Wikipedia, can be imported to facilitate the analysis. The type prediction approach described here is able to improve the state-of-the-art in this task by injecting the social media data as additional features.

Secondly, a concealing approach was presented to hide true user's interests from the user profiling pipelines. The approach exploits the LOD's categorical knowledge, acquired using SocialLink as a bridge, to first predict latent user's interests and then propose actions to perform by the user so that the interests can no longer be reliably predicted. This is done by finding the optimal configuration of user's followed accounts making the resulting distribution over interests, as inferred by the user profiling system, as close as possible to uniform. This approach is a vivid example of a novel task enabled by the knowledge transfer towards social media.

Finally, the Social Media Toolkit (SMT) system was developed as part of the project providing additional functionality over SocialLink. SMT, among other things, uses SocialLink to implement two distinct Named Entity Linking (NEL) scenarios. Firstly, SMT is able to perform direct disambiguation of user mentions in social posts against a knowledge base using the SocialLink dataset. This functionality allows improving the performance of typical NEL pipelines not only by directly solving the task for some of the mentions but by also providing context for the rest of the target text. The MicroNeel system was developed to showcase this scenario reaching the second place in the EVALITA2016 entity linking competition.

In the second scenario, SMT is able to perform NEL on arbitrary text against the social media using SocialLink's linking pipeline. Here the NEL task is modified: instead of linking each mention to a corresponding entity in a KB, a linking to a suitable social media profile is performed. As a clear demonstration of the latter scenario, SMT was incorporated into the Social Media Management platform called Pokedem. Pokedem is designed to recommend specific actions to perform on Twitter in order to increase the

popularity of the managed account. The NEL capabilities of SMT are used in Pokedem to augment the proposed tweet with mentions of social media users producing richer content.

To summarize, in this thesis I discuss the knowledge transfer between DBpedia and Twitter, additionally detailing use cases and approaches that are enabled by this transfer in both directions (graphically depicted in Figure 1.1). The major contributions of this thesis, along with their relevant publications and the chapters covering them are as follows:

**Contribution C1 – SocialLink Approach** An automatic pipeline designed to link LOD-compliant Knowledge Bases to social media profiles (Nechaev et al. 2017b; Nechaev et al. 2018b, Chapter 3).

**Contribution C2 – SocialLink Resource** The LOD dataset linking 2.5M entities from DBpedia to the corresponding Twitter accounts (Nechaev et al. 2017d, Chapter 4).

**Contribution C3 – Type Prediction** A type prediction approach employing social media data to consistently outperform state-of-the-art systems (Nechaev et al. 2018c, Chapter 5).

**Contribution C4 – Concealing User Interests** A system designed to protect interests of social media users from being inferred by the typical user profiling approaches (Nechaev et al. 2017a, Chapter 6)

**Contribution C5 – Social Media Toolkit** A system providing two novel Named Entity Linking approaches to take advantage of social media data (Corcoglioniti et al., 2016, 2017, 2018, Chapter 7)

## 1.4 Structure of the Thesis

The remainder of the thesis is structured as follows:

**Chapter 2** provides needed background including approaches, techniques, systems and other related work used throughout this thesis. Related work, that is specific for individual contributions, is provided in the end of the respective chapters.

**Chapter 3** presents the SocialLink approach (Contribution **C1**). Here I present the linking task in details, identify challenges and go through all the steps of the designed solution. Additionally, I provide the extensive evaluation of the approach along with the thorough error analysis and the discussion on approach limitations.

**Chapter 4** describes the SocialLink resource (Contribution **C2**). This chapter presents the resource design aspects, core statistics, formats and provides necessary details for recreating the resource.

The following chapters describe systems and approaches based on SocialLink together covering the rest of the contributions of this thesis:

**Chapter 5** describes the state-of-the-art type prediction system exploiting social media data to provide predictions for eight DBpedia types. I provide extensive evaluation showcasing the performance of the social media data extracted exploiting links from DBpedia to Twitter. This chapter covers Contribution **C3**.

**Chapter 6** discusses the usage of the LOD data for the user profiling task in social media. Specifically, I tackle the task of interests prediction of passive users. As a main contribution, I present the task of concealing user interests and describe the approaches developed to solve it. This chapter represents Contribution **C4**.

**Chapter 7** details the Social Media Toolkit system and it's features. Additionally, this chapter covers the two systems built using the SMT capabilities: MicroNeel and Pokedem. This chapter covers Contribution **C5**.

Finally, I conclude this thesis with the following chapter:

**Chapter 8** summarizes the thesis results and provides extensive discussion on the potential uses, limitations, future improvements for the work presented in previous chapters.

## 1.5 Publications

The core publications supporting the main contributions (**C1**, **C2**) of this thesis are listed below:

- Nechaev, Y., Corcoglioniti, F., and Giuliano, C. (2017b). Linking Knowledge Bases to Social Media Profiles. In *Proceedings of the Symposium on Applied Computing, SAC 2017, Marrakech, Morocco, April 3-7, 2017*, pages 145–150
- Nechaev, Y., Corcoglioniti, F., and Giuliano, C. (2017d). SocialLink: Linking DBpedia Entities to Corresponding Twitter Accounts. In *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II*, pages 165–174
- Nechaev, Y., Corcoglioniti, F., and Giuliano, C. (2018b). SocialLink: Exploiting Graph Embeddings to Link DBpedia Entities to Twitter Profiles. *Progress in AI*, 7(4):251–272

Additional publications supporting the rest of the contributions are as follows:

- Nechaev, Y., Corcoglioniti, F., and Giuliano, C. (2018c). Type Prediction Combining Linked Open Data and Social Media. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 1033–1042

- Nechaev, Y., Corcoglioniti, F., and Giuliano, C. (2017a). Concealing Interests of Passive Users in Social Media. In *Proceedings of the Re-coding Black Mirror 2017 Workshop co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 22, 2017.*

- Corcoglioniti, F., Giuliano, C., Nechaev, Y., and Zanoli, R. (2017). Pokedem: An Automatic Social Media Management Application. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, pages 358–359, New York, NY, USA. ACM

- Corcoglioniti, F., Nechaev, Y., Giuliano, C., and Zanoli, R. (2018). Twitter User Recommendation for Gaining Followers. In *AI*IA 2018 Advances in Artificial Intelligence - 17th International Conference of the Italian Association for Artificial Intelligence, Trento, Italy, November 20-23, 2018, Proceedings*

- Corcoglioniti, F., Aprosio, A. P., Nechaev, Y., and Giuliano, C. (2016). MicroNeel: Combining NLP Tools to Perform Named Entity Detection and Linking on Microposts. In *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016), Napoli, Italy, December 5-7, 2016.*

## 1.6 Artifacts

During the development of the approaches and systems detailed here, multiple artifacts were produced ranging from software repositories, such as our main SocialLink project repository, to resources and supplementary materials supporting the publications related to this thesis. They are as follows:

- SocialLink project including SMT – `https://github.com/Remper/sociallink`
- Type Prediction approach – `https://w3id.org/sociallink/type-prediction`
- Concealing user interests approach – `https://github.com/Remper/re-coding-ws`
- Training and model generation code for Pokedem – `https://github.com/Remper/pokedem-models`
- Twitter knowledge extraction pipelines – `https://github.com/Remper/tweetframe`

- SocialLink resource – https://w3id.org/sociallink#download
- SocialLink gold standard – https://w3id.org/sociallink#download

# Chapter 2

# Background

In this chapter, I start by briefly introducing the relevant Semantic Web concepts, such as the Linked Open Data, most prominent datasets, related technologies and tools that are employed in this thesis. Then, I will highlight the key concepts of the social media. Finally, the majority of this chapter is dedicated to outlining some of the fundamental principles and state-of-the-art approaches employed to represent textual and graph data that are used throughout this thesis. In addition to the content presented in this chapter, going forward I will also provide a more in-depth look at related work and relevant background in each chapter separately, where that information pertains specifically to the contents discussed in the chapter.

## 2.1 Semantic Web and Linked Open Data

Semantic Web is the initiative that aims to extend the World Wide Web with the goal of facilitating the rapid sharing and reuse of data. This goal is achieved by developing open standards and best practices under the World Wide Web Consortium (W3C)[1] making data in the web accessible, machine-readable and increasingly interlinked.

A centerpiece of the Semantic Web initiative is the Linked Open Data (LOD) cloud, which is a collection of interlinked datasets that are released under an open license. A LOD-compliant dataset has to adhere to a list of simple rules: (i) the instances described in the dataset have to be identifiable via URIs, (ii) when queried, descriptions have to be returned using open standards, such as RDF and SPARQL, and (iii) links have to be provided to other related URIs in other datasets from the LOD. Such principles were declared essential for the success of the Semantic Web project (Berners-Lee, 2006) and direct the community effort to share as many interlinked datasets as possible making all sorts of knowledge publicly available in a standardized machine-readable form. This effort has yielded billions of RDF triples accessible from thousands of different datasets over

[1] http://www.w3.org

the last years. With such vast array of knowledge readily available, the LOD cloud is routinely used as a source of background knowledge for tasks spanning across many fields, for example, in Computer Vision and Natural Language Processing.

Independently from the LOD concept, Wilkinson et al. (2016) have defined a set of principles, namely, Findability, Accessibility, Interoperability, Reusability (collectively known as FAIR principles) aimed at supporting the reuse of data produced by researchers. It has since been adopted by academia, including the major Semantic Web outlets, such as ISWC and ESWC conferences, and industry alike as guidelines to consider when building and sharing datasets to improve data discovery, access, integration, adequate citation, and reuse. One of the major contributions of this thesis, the SocialLink resource, is build both to be LOD-compliant and having FAIR principles in mind.

One of the largest datasets existing in the LOD cloud is DBpedia (Lehmann et al., 2015). DBpedia is built by automatically extracting structured multilingual knowledge from Wikipedia, the crowdsourced online encyclopedia, meaning that each entity contained in DBpedia is backed by the corresponding Wikipedia article. In total, DBpedia has 128 language chapters covering corresponding Wikipedia ones and provides over 1.46[2] billion RDF triples. Being the most interlinked dataset in the LOD cloud, it acts as a centerpiece providing a simple access point for the researchers and industry alike to the world of linked data.

On the other hand, the Wikidata[3] project of the Wikimedia Foundation is a collaborative effort aiming at creating a knowledge backbone for the Wikipedia and related projects accumulating structured knowledge in a LOD-compliant form. Wikipedia is able to populate its infoboxes and other structured parts from the knowledge contained in Wikidata. Much like DBpedia, Wikidata is multilingual and each Wikipedia article has its Wikidata counterpart entity. Following best LOD and FAIR practices, it is freely accessible and interlinked with various datasets within and outside the LOD cloud. Additionally, Wikidata aims at providing references for the claims it contains, allowing the user to verify the knowledge by checking where it comes from. Unfortunately, many of the claims in Wikidata are not appropriately sourced making SocialLink, which is a core contribution of this thesis, even more relevant as it allows using social media profiles as a source of references.[4] Additionally, even though SocialLink mainly targets DBpedia, Wikidata URIs are used where possible in the resource to identify entities linking them to DBpedia URIs via the owl:sameAs links.

LOD-compliant datasets typically rely on Resource Description Framework (RDF) specification to represent its knowledge. RDF 1.1 (Wood et al., 2014) presents data

[2] As described in Lehmann et al. (2015)
[3] https://www.wikidata.org
[4] See the soweego project discussed in Section 4.4.4.

as subject-predicate-object triples forming a knowledge graph. A node in such graph (a subject or an object of a triple) can either be a URI, a typed Unicode string literal or a blank node (denoting anonymous resources), while the predicate has to be a URI representing a relationship. Type information for arbitrary nodes can be expressed using the predicate rdf:type together with object URIs identifying types. The semantics of these predicates and types may be described using ontological languages such as RDFS (Guha and Brickley, 2014) and OWL (Motik et al., 2012), themselves expressible in RDF. In this thesis, I use OWL 2 to formalize the semantics of SocialLink data. RDF graphs can be serialized in many formats, such as Turtle, RDF/XML, JSON-LD, RDF/JSON, and many others. To perform various manipulations with RDF knowledge graphs, such as statistics extraction, smushing, filtering, deduplication, and other transformations, many tools have been developed over the last years. In this thesis, unless stated otherwise, I primarily employ the RDFpro (Corcoglioniti et al., 2015) tool to perform the large-scale local processing of RDF data. RDFpro is an open source Java library with a command line interface that implements many of the operations typically needed to produce and work with LOD datasets.

In order to query and manipulate the RDF-based knowledge graphs, the SPARQL Protocol and RDF Query Language (SPARQL) was introduced and standardized by the W3C (Harris and Seaborne, 2013). SPARQL endpoints offer access to datasets using a simple HTTP-based protocol supporting multiple output formats. The rich query language of SPARQL resembles other querying languages, such as SQL. It offers read queries that can be restricted with WHERE clauses to define patterns and constrain the query; then a variety of standard operations, such as joins, unions, intersections, aggregations, subqueries, and others exist to further transform query results. SPARQL also supports update queries that modify stored RDF triples. Additionally, the latest specification adds support for federated queries that allow accessing RDF data distributed across multiple endpoints. SPARQL has a number of open source implementations including RDF4J, Jena[5] and Virtuoso[6], GraphDB[7] and BlazeGraph.[8] In this thesis, I employ the KnowledgeStore[9] storage system with a Virtuoso backend to store, manage and query the RDF-based datasets. KnowledgeStore provides a SPARQL endpoint and offers a web-based user interface to manage the stored RDF triples and associated metadata.

[5] https://jena.apache.org
[6] https://virtuoso.openlinksw.com
[7] http://graphdb.ontotext.com
[8] https://www.blazegraph.com
[9] https://knowledgestore.fbk.eu

## 2.2 Social Media

Social media has attracted a considerable amount of research attention over the last decade due to its popularity and its provision of enormous amounts of real-time knowledge. Social media are mostly proprietary outlets that provide an opportunity for the users to create and post content, to search and interact with other users and the content created by them. Users create and fill their public or private profiles, which may or may not include machine-readable attributes describing them. Generally speaking, the users cannot verify if the profile they are interacting with is genuine, making it easy for an impostor to steal someone else's identity or invent an entirely new one. However, the social network can in some cases verify the identity of high-profile public entities partially alleviating this issue. The current major social networks include Facebook with $2.3B$ (statista.com, 2019a) monthly active users, Instagram with $1B$ (statista.com, 2019b) and Twitter with over $300M$ (statista.com, 2019c).

Not every social network, however, is proprietary and closed source. Since the publication of the OStatus[10] standard and its successor, ActivityPub,[11] by the W3C, there has been an increased community effort to create a new generation of social media based entirely on open standards increasing the end users' ability to control their data and improving compatibility between different social media. The most prominent examples include Mastodon[12] and GNU Social.[13]

The key concept of any social media is the social graph — a directed graph of users that captures explicit and implicit interactions between them. It may be implemented via some kind of friend, follow or subscribe action that generally puts the content written by one user to the other users content feed. However, even if the user does not explicitly state their desire to subscribe to someone, the mere fact of reaction to someone else's content (via like, share or comment functionality) may be enough to establish the link between two users. As will be repeatedly shown throughout this thesis, the social graph can reveal much and more about any given user.

Many social media provide APIs for third parties to access a subset of the available users' data. This data may include the above-mentioned social graph, user's public profile, user's public content and some metadata, such as location and event data, profile settings and popularity statistics. Even though the released data is typically incomplete due to privacy or business reasons, it has been repeatedly shown[14] that a third party can exploit

[10] https://www.w3.org/community/ostatus
[11] https://www.w3.org/TR/activitypub
[12] https://mastodon.social
[13] https://gnu.io/social
[14] See, for example, Section 6.2 for the review of the user profiling task that specifically tackles this problem.

this data to fill out the blanks in the user's profile and gather the information that was not supposed to be available, making the privacy implications of the social media an increasingly important topic of public discussion. In case the API is not present, social media data can be collected using a web crawler, which, however, can in many cases violate the terms of use.

Social media are being routinely used in the academic community to assist in solving a wide variety of tasks spanning across many fields. I will provide a review of the tasks that are relevant to this thesis, namely user profiling (Section 6.2), and profile matching (Section 3.8), in the subsequent chapters.

## 2.3 Representation Learning

The performance of machine learning approaches significantly depends on the way we represent input data. Typically, for each data type, there is a set of conventional approaches to extract features from the raw input. For example, categorical input is usually represented as a sparse vector $\mathbf{x} \in \mathbb{R}^l$, where $l$ is the number of possible values for this particular input. Numeric features are normalized in some way to range from $-1$ to 1 or from 0 to 1 depending on the approach. Specialized representations exist for more complex data types, such as text, images, graphs and other.

Additionally, representations for a particular set of features can be learned (see Chapter 15 in Goodfellow et al. 2016). The idea is to acquire a representation for a given set of features as a by-product of one learning algorithm and then use it as input for another algorithm, effectively sharing the learned knowledge across multiple tasks. In particular, suppose that we set up a feed-forward neural network that is trained to solve task $A$ given some representation of an object as input. Regardless of the chosen initial feature set, by the final layer, the feed forward neural network will naturally learn to represent this object most suitably based on the task at hand. For example, if the classes in the original problem were not linearly separable given the input features, they may become separable by the final hidden layer. We can then use this representation at the last hidden layer and feed it as input to task $B$ that expects features based on this object as input. In this case, we are effectively collecting what we have learned about the object at hand during solving the first task and reusing it (transferring this learning) in a different task.

The concept of learning representations becomes even more important considering that we might have plenty of data to solve task A but not enough to solve task B even if a researcher tries hard to extract the best possible features from the input object. In real scenarios, we would typically have a significant number of unlabeled samples and a relatively limited number of labeled ones. In this case, we would usually learn representations in an unsupervised way by designing an unsupervised objective to highlight

certain inherent properties of objects in a dataset. For example, for the text we might follow the *distributional semantics* hypothesis described below or corrupting the input sequence in some way and training the approach to correct the mistake making the learning algorithm to encode the meaning of the sequence into its internal state and then decoding the corrected version.

In the subsequent sections, I will describe the core representation learning techniques that I employed in this thesis. Tasks tackled here routinely use the notion of a social media user or an knowledge base entity as input for both of which a vast amount of unsupervised knowledge is available. Given that the size of a training set for each individual task is rather small to learn representations for such complex objects from the ground up, the idea of learning them using an unsupervised approach[15] is of paramount importance.

## 2.4 Text Representations

In this section, I briefly recap the algorithms and techniques used to represent written text in vectorial form throughout this thesis. Textual features are a cornerstone of any analysis done on social media and provide significant performance improvements for many of the tasks in the Semantic Web.

The most straightforward way to represent written text is a bag-of-words model. There, the input text is converted to a sparse vector $\mathbf{x} \in \mathbb{R}^v$ where $v$ is the size of the vocabulary. Each non-zero index of this vector contains a score associated with a specific term that is present in the text. In the simplest case, each term in a text has a score of 1.0, yielding a so-called one-hot representation. In general, the score consists of two components: term frequency (TF) and the inverse document frequency (IDF). Semantically, TF is a frequency with which the term appears in the input text, while the IDF is the inverse frequency with which the term appears in the corpus. While there exist many weighting schemas for both TF and IDF, in this thesis I will employ the following formula for each element of $\mathbf{x}$:

$$x_t = \text{tf}(t) \cdot \text{idf}(t, D) = \log(1 + \text{freq}_t) \cdot \log\left(1 + \frac{|D|}{1 + |\{d \in D : t \in d\}|}\right)$$

where $t$ is a term, $d$ is a document in a corpus (e.g., a tweet in a corpus of tweets), $D$ is a chosen corpus and $\text{freq}_t$ is the number of occurrences of $t$ in the document.

Such representation, however, is suboptimal for many applications (see Section 6 in Zhang et al. 2016). A high-dimensional sparse representation resulting from the bag-of-words approach is increasingly difficult to use as input with many of the machine learning

[15] This process usually referred to as *unsupervised pretraining*

algorithms, especially the ones based on neural networks, as each word index has to be associated with an individual weight and sparse tensor operations are much less performant than their dense counterparts on modern hardware.

Additionally, the bag-of-words model suffers from a *vocabulary mismatch* problem. The *vocabulary mismatch* arises when there is a need to group and compare semantically similar items. In a bag-of-words model, "cat" and "tiger" yield orthogonal representations even though both terms refer to felines. It means that "cat" would be as distant from "tiger" as it is from, let's say, "table", providing no extra semantic information besides a simple token match to the downstream models.

In order to reduce the dimensionality of the input while also solving the *vocabulary mismatch* issue, various algorithms were proposed. In these approaches, each term $i$ is encoded into a dense low-dimensional continuous representation $\mathbf{w}_i \in \mathbb{R}^d$, where $d$ is a chosen size of the representation, commonly referred to as *word embedding*. The word embeddings are typically acquired using an unsupervised algorithm that uses a large corpora to learn representations according to some hypothesis. Modern embedding algorithms used in practise, such as LSA, word2vec and GloVe, employ a so-called *distributional semantics* hypothesis which states that words that appear in similar contexts tend to have similar meanings. Given that we are interested in learning a vector for each word, we would like the words that appear in similar contexts to be close to each other in the resulting vector space according to some similarity metric, such as *cosine similarity*.[16] As can be seen, the vocabulary mismatch problem, that made all words orthogonal to each other, is solved since the words in a new vector space are arranged based on their meaning. Note that here and onwards I would use "word", "term" and "token" as synonyms, while it is assumed that in a text we can encounter other tokens such as punctuation, numbers and even emoji. Once the word embeddings are populated, dense low-dimensional representation of text can be acquired by simply multiplying the sparse TF-IDF vector $\mathbf{x}$ by the embedding matrix $\mathbf{M}$:

$$\mathbf{x}_{\text{dense}} = \mathbf{x}^T \cdot \mathbf{M} \tag{2.1}$$

Throughout this thesis, I employ various embedding algorithms from the literature that follow the distributional semantics hypothesis. Here, I describe three of them, LSA, GloVe and Swivel, in details as they are used in the proposed approaches and briefly discuss word2vec and fastText.

[16] $\text{sim}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}$

### 2.4.1 LSA

Latent Semantic Analysis (LSA, Deerwester et al. 1990) is a technique that allows learning of word representations from the co-occurrence matrix of words and documents. Given a corpus of $n$ documents having a vocabulary of $v$ unique words, we can build a co-occurrence matrix $\mathbf{X} \in \mathbb{R}^{v \times n}$ such that $x_{ij}$ corresponds to some frequency measure (which can be, for example, the above-mentioned TF-IDF weight) of $i$th word of the vocabulary in $j$th document. From such matrix we can already derive word vectors of size $n$ by taking the corresponding rows of $\mathbf{X}$. The resulting vector space will already be arranged according to distributional semantics and we can compute similarity score between the words using those vectors.

However, the original matrix $\mathbf{X}$ is large, noisy and sparse. To reduce the dimensionality of $\mathbf{X}$, LSA proposes to employ singular value decomposition (SVD), splitting $\mathbf{X}$ into a product of two orthogonal matrices and a diagonal matrix: $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \mathbf{U} \in \mathbb{R}^{v \times v}, \mathbf{\Sigma} \in \mathbb{R}^{v \times n}, \mathbf{V} \in \mathbb{R}^{n \times n}$. By taking the $d$ largest singular values from $\mathbf{\Sigma}$ and their corresponding singular vectors from $\mathbf{U}$ and $\mathbf{V}$, we acquire an approximation of $\mathbf{X}$: $\mathbf{X}_d = \mathbf{U}_d\mathbf{\Sigma}_d\mathbf{V}_d^T, \mathbf{U}_d \in \mathbb{R}^{v \times d}, \mathbf{\Sigma}_d \in \mathbb{R}^{d \times d}, \mathbf{V}_d \in \mathbb{R}^{d \times n}$. This factorization procedure is referred to as *truncated SVD* and it is shown that $\mathbf{X}_d$ is the closest possible approximation of $\mathbf{X}$ with rank $d$ matrices. According to the LSA procedure, if we then compute:

$$\mathbf{M}_{\text{lsa}} = \mathbf{U}_d \cdot \mathbf{\Sigma}_d, \quad \mathbf{M}_{\text{lsa}} \in \mathbb{R}^{v \times d} \tag{2.2}$$

the rows of $\mathbf{M}_{\text{lsa}}$ can be employed as embeddings of size $d$ for the corresponding tokens yielding a convenient low-dimensional representation based on unsupervised co-occurrence statistics derived from the selected corpus. In this thesis, we employ the LSA model built basen on the corpus derived from seven language chapters of Wikipedia and detailed in Aprosio et al. (2013). I use this model to represent textual content in both Twitter and DBpedia across all the approaches presented in this thesis.

### 2.4.2 GloVe and Swivel

Recently proposed GloVe (Pennington et al., 2014) and Swivel (Shazeer et al., 2016) are the global log-bilinear regression models that implement the distributional hypothesis to learn word vectors. The co-occurrence matrix used in these methods is term-term instead of term-document, meaning that $x_{ij}$ contains the frequency with which the term $i$ appears in the context of the term $j$ yielding the square matrix $\mathbf{X} \in \mathbb{R}^{v \times v}$, where $v$ is the size of vocabulary. The context of any given word consists of all the words around it up to a certain distance. So, given a sentence $S = \{t_1, t_2, t_3, t_4, t_5\}$ and distance $l = 2$, $t_1$ would be co-occurring with $t_2$ and $t_3$, while $t_4$ would be co-occurring with $t_2$, $t_3$ and

$t_5$. Such definition of a context was initially proposed in Lund and Burgess (1996) for HAL approach. Typically, the weight of a single co-occurrence between any two words is defined as $1/l$ given that they are $l$ words apart, which are then summed up over all co-occurrences of those two words in the corpus.

After the co-occurrence statistics are computed, the approximate factorization of $\mathbf{X}$ into matrices $\mathbf{W} \in \mathbb{R}^{v \times d}$ and $\tilde{\mathbf{W}} \in \mathbb{R}^{v \times d}$ is performed by formulating an optimization problem and minimizing the appropriate objective function treating $\mathbf{W}$ and $\tilde{\mathbf{W}}$ as weights. To do that GloVe defines the following objective function with the goal of encoraging $\mathbf{W}\tilde{\mathbf{W}}$ to approximate $\log(\mathbf{X})$:

$$J_{\text{GloVe}} = \sum_{i,j} f(x_{ij})(\mathbf{w}_i^\top \tilde{\mathbf{w}}_j + b_i + \tilde{b}_j - \log x_{ij})^2 \tag{2.3}$$

where $f(\cdot)$ is a weighting function and $b_i$ and $\tilde{b}_j$ are the biases. The weighting function is selected in such way so that the frequent co-occurrences between words are not overweighted. Additionally, $f$ is chosen so that $f(0) = 0$, meaning that sparse values in $\mathbf{X}$ (which usually constitute 75-95% of values in such matrix) can be skipped during training which significantly improves the performance of the approach for large $v$. Finally, one of the resulting matrices $\mathbf{W}$ or $\tilde{\mathbf{W}}$ can be defined as a matrix $\mathbf{M}_{\text{GloVe}}$ of word embeddings of size $d$, which can be used in eq. 2.1 to acquire a dense representation of a text.

Swivel builds on the GloVe approach but instead of approximating $log(\mathbf{X})$ it estimates the *pointwise mutual information* (Church and Hanks, 1990) between the terms $\text{pmi}(i; j)$, which is defined as follows:

$$\text{pmi}(i; j) = \log \frac{P(i, j)}{P(i)P(j)} = \log \frac{x_{ij}|D|}{x_{i*}x_{*j}} \tag{2.4}$$

where $x_{i*} = \sum_j x_{ij}$, $x_{*j} = \sum_i x_{ij}$ and $|D| = \sum_{i,j} x_{ij}$. Which modifies the objective function (eq. 2.3) as follows:

$$\begin{aligned} J_{\text{swivel}} &= \frac{1}{2} \sum_{i,j} f(X_{ij})(\mathbf{w}_i^\top \tilde{\mathbf{w}}_j + b_i + \tilde{b}_j - \text{pmi}(i; j)))^2 \\ &= \frac{1}{2} \sum_{i,j} f(X_{ij})(\mathbf{w}_i^\top \tilde{\mathbf{w}}_j + b_i + \tilde{b}_j - \log x_{ij} - \log |D| + \log x_{i*} + \log x_{*j})^2 \end{aligned} \tag{2.5}$$

The primary distinction between Swivel and GloVe is the introduction of the "soft hinge" loss for cases where $x_{ij} = 0$:

$$J_{\text{hinge}} = \log[1 + \exp(\mathbf{w}_i^\top \tilde{\mathbf{w}}_j - \log |D| + \log x_{i*} + \log x_{*j})] \tag{2.6}$$

This special case allows the unobserved co-occurrences to contribute to the learning process, producing more stable estimates for rare words. It is worth noting, however, that the introduction of the special case for zero values of matrix $\mathbf{X}$ negatively affects performance making it impractical to use Swivel for learning embedding matrices with large vocabularies. To mitigate this issue, the authors of Swivel propose to split the co-occurrence matrix into shards making the whole training step easily parallelizable. Even though, Swivel and GloVe were originally designed to represent words, it is shown (Pennington et al., 2014; Nechaev et al., 2018b; Cochez et al., 2017b) that the same log-bilinear models can be exploited for any type of objects — it is just the matter of defining and computing the appropriate co-occurrence matrix. In particular, these approaches will be used in this thesis to represent nodes in a graph.

## 2.5 Graph Embeddings

Recently, learning of dense low-dimensional representations for nodes in a graph (onwards *graph embeddings*) to provide a better alternative to sparse representations have become a hot topic in the community (Cai et al., 2018). Such representations are typically learned using an unsupervised approach employing the entire graph for training. The task is similar to the one of learning word embeddings in NLP: given the graph $G = (V, E)$, where $V$ are the nodes and $(v_i, v_j) \in E$ are the edges, we would like to learn an embedding matrix $\mathbf{X}_G \in \mathbb{R}^{|V| \times d}$ so that rows give $d$-dimensional representations for each node in a shared vector space. Alternatively, if the edges can be typed (labeled), the graph definition changes to $G = (V, E, R)$ with a set of possible relation types $R$ and triples representing edges $(v_i, r_k, v_j) \in E, r_k \in R$. In this case representations can also be learned for the relation types themselves (Nickel et al., 2011; Guu et al., 2015; Bordes et al., 2013; Socher et al., 2013). However, even when the type information is available (e.g., knowledge graphs in the LOD), I will only employ representations for nodes for tasks presented in this thesis. Here, I provide the review of the approaches employed over the last years to acquire graph embeddings including the ones that are either directly used (Cochez et al., 2017b; Ristoski et al., 2017) in this thesis or that influenced (Tang et al., 2015; Nickel et al., 2011) the design decisions made in Chapter 3 to produce embeddings for Twitter users derived from the social graph.

### 2.5.1 General Graph Embedding Methods

The importance of representing graph-based features in the context of this thesis is hard to overstate. In the Semantic Web community, given that any LOD dataset is essentially a knowledge graph, graph-based features play a key role in any machine learning-based

task there. In social media analysis, the usage of social graphs enables tasks that wouldn't be possible otherwise (e.g., see interests prediction for passive users in Chapter 6) and significantly improves performance of approaches for tasks like user profiling, profile matching, recommendation and event detection.

After the overwhelming success of learning embeddings to represent text in Natural Language Processing (see Section 2.4), researchers have started to try to use the same ideas, namely the distributional hypothesis, to learn graph embeddings. In such works, the graph is seen as a document, while the nodes are viewed as tokens in this document. However, documents have a linear structure, meaning that the sequence of tokens can be scanned sequentially and each word has a clear notion of context, while the graphs do not naturally posses this property. In order to resolve this issue, the graph has to be linearized in some way to resemble a sequence from which the context for each node can be derived.

DeepWalk (Perozzi et al., 2014) approach was one of the first to linearize the graph to learn embeddings exploiting the methods designed to work on text. Its authors proposed to use uniform random walks on graph to model it as a set of sequences to run a Skipgram model of word2vec on them. Their experiments on multiple datasets showed the effectiveness of the approach and pawed the way for other researchers to further explore this direction. One of the extensions of the DeepWalk was the Deep Graph Kernels (Yanardag and Vishwanathan, 2015) system that learned embeddings for structured objects, such as graphlets and strings instead of nodes. The authors of DGK have also explored a number of alternative linearization mechanisms other than random walks. Direct usage of the Skipgram objective is not the only way to learn graph embeddings. Instead, LINE (Tang et al., 2015) proposes to directly model the first and second order proximity using *breadth-first search* starting from the sampled node. LINE approach is conceptually similar to matrix factorization methods but is designed to preserve the original network structure instead of just node similarity.

Node2vec (Grover and Leskovec, 2016) builds upon and generalizes the DeepWalk approach. Node2vec authors further formalize the random walk generation procedure by introducing two parameters. The return parameter $p$ controls the probability with which the random walk procedure is able to use the edge it had just traversed and the in-out parameter $q$ controls the probability of leaving or staying in the tightly connected groups of nodes. Different configurations of those two parameters control the way with which the random walking procedure balances *breadth-first search* and the *depth-first search*. Authors note that DeepWalk is effectively a special case of node2vec with $p = 1$ and $q = 1$. The additional flexibility in controling the linearization procedure allowed Node2vec to outperform both the LINE and DeepWalk approaches.

Additionally, approaches designed to learn embeddings for both nodes and relations in knowledge graphs specifically have been explored. RESCAL (Nickel et al., 2011) factorize

the tensor of relations and nodes learning representations for both as a result. One of the possible variations of this model was proposed by Guu et al. (2015). HolE (Nickel et al., 2016) offers a more efficient yet a more expressive version of RESCAL by simplifying the model and the objective via the usage of circular correlation operator to obtain the composition of embeddings. Similar compositional approach but in a pure neural network-based scenario was also proposed in Socher et al. (2013) (Neural Tensor Networks). There they learn low-dimensional representations for entities and a combination of a rank 2 and rank 3 tensors to represent relation. TransE (Bordes et al., 2013), on the other hand, learns representations for entities and relations by contrasting the real subject-relation-object triples with the corrupted ones assigning the loss penalty if the corrupted triple is scored higher than the correct one. TransE has a multitude of extensions including TransH (Wang et al., 2014) and TransR (Lin et al., 2015) that aimed to solve various limitations of the original approach. TransE has been routinely used in the last years for the *link prediction* task, which is a generalized version of the *type prediction* task explored in Chapter 5 of this thesis.

### 2.5.2 RDF Embeddings

LOD knowledge bases consist of RDF triples subject-relation-object, which can naturaly be represented as a graph with typed edges defined above. While any of the approaches described in Section 2.5 can be used to represent nodes in such graphs, many researchers (Ristoski et al., 2017; Ristoski and Paulheim, 2016a; Cochez et al., 2017a,b) have tried to tune their approaches for RDF-based knowledge graphs specifically resulting in improved performance for downstream tasks in the LOD. I will refer to embeddings produced by such specialized approaches as RDF embeddings.

One of the approaches for training RDF embeddings, RDF2Vec (Ristoski et al., 2017; Ristoski and Paulheim, 2016a), uses RDF graph kernels from the recent literature, specifically the Weisfeiler-Lehman Subtree RDF graph kernels (de Vries, 2013; de Vries and de Rooij, 2015), to perform linearization of nodes in a graph. This combined with the breadth-first search random walks from DeepWalk (Perozzi et al., 2014) allowed them to train embeddings using the usual Skipgram objective. The two linearization strategies are complementary meaning that one or the other can be skipped based, for example, on the task performance requirements. RDF2Vec authors have provided pretrained embeddings for DBpedia and Wikidata allowing other researchers to employ them in downstream tasks without reimplementing the approach or even retraining. The resulting embeddings semantically group entities of a KB. For example, dbr:France have dbr:Germany and dbr:Italy as nearest neighbours, while dbr:Paris is close to dbr:Berlin and dbr:Rome. Additionally, we can see that the chosen linearization procedure preserves other properties that can

typically be found in distributional semantics-based language models: linear substructures that capture relations between words in methods, such as GloVe, can also be found in embeddings produced by RDF2Vec. For example, if we substract dbr:Germany vector from dbr:Berlin and then add dbr:Italy, the nearest entity would be dbr:Rome meaning that for such entity the property dbo:Country and the complementary property dbo:Capital are implicitly preserved.

While the RDF2Vec approach focuses on using Skipgram objective to learn RDF graph embeddings for DBpedia and Wikidata, the subsequent work by Cochez et al. (2017b) employs the GloVe approach I described in Section 2.4.2. The usage of a method based on factorization of the co-occurrence matrix allowed the authors to explore approaches that do not rely on linearizing a graph into a set of sequences. Instead, reliance on the co-occurrence matrix only requires some notion of importance to be computed between a target and any other node in order for the approach to function. Cochez et al. (2017b) employ the Personalized PageRank (PPR) (Page et al., 1999) algorithm to produce such importance statistics. However, since the PPR had to be computed for each node separately and the number of nodes is high in the target knowledge graphs, the authors opted to use the Bookmark-Coloring Algorithm (BCA) (Berkhin, 2006) to produce an approximation of PPR. The authors further improve the performance of this method by reusing many of the values for the previously computed nodes.

Finally, the BCA approach can take into account edge weights. As DBpedia does not have a readily available notion of weight, Cochez et al. (2017b) propose twelve different weighting schemas based on raw frequences (both edge-centric and node-centric), different variations of Wikipedia-based PageRank and uniform weights. Such weighting schemas for RDF graphs were originally introduced in a prior work by the same authors (Cochez et al., 2017a) and applied as an extension to the original RDF2Vec. Throughout this thesis, I will employ the precomputed versions of their RDF embeddings with different weighting schemas to represent knowledge graph nodes in DBpedia and Wikidata.

## 2.6 Conclusions

Here I have presented relevant background I will refer to throughout this thesis: basic Semantic Web and Social Media concepts; the brief review of the state-of-the-art data representation approaches including the ones used to represent text and various graph data. As the subsequent chapters cover different topics, additionally I will provide the review of those topics in the corresponding sections. Specifically, Section 3.8 provides relevant background for the linking task that is at the core of this thesis including the review of the profile matching task and some of the similar linking techniques used in social media analysis. Section 5.7 talks about recent type prediction approaches and feature extraction

from social media content. Section 6.2 covers the user profiling task in general with the emphasis on interests inference, privacy issues, and binarized neural networks. Finally, Section 7.3.1 talks about follower acquisition strategies and recommendation techniques.

# Chapter 3

# SocialLink Approach: Linking Knowledge Bases to Social Media Profiles

The core contribution of this thesis is SocialLink: a machine learning-based approach and the accompanying LOD-compliant dataset for linking social media profiles to corresponding entities in the knowledge base. Built to bridge the vibrant Twitter social media world and the Linked Open Data cloud, SocialLink enables knowledge transfer between the two, both assisting Semantic Web practitioners in better harvesting the vast amounts of information available on Twitter and allowing leveraging of DBpedia data for social media analysis tasks.

In this chapter, we describe the centerpiece of SocialLink: its linking approach. First, we introduce the base three-phase procedure exploiting readily available features. Secondly, we discuss the addition of graph features from both the LOD and social media sides into the pipeline. To successfully utilize the new features, we propose a redesigned deep neural network-based candidate selection algorithm. Finally, we present the extensive evaluation of both the original and the redesigned versions demonstrating high performances on the gold standard dataset.

**Acknowledgments** The thesis includes material resulting from collaborative work or described in previous publications, because of that from this point on up until the Conclusions chapter, I will use "we" and "us" pronouns as an acknowledgment to my co-authors.

## 3.1 Introduction

Social media is popular among organizations, celebrities, politicians and other public entities allowing them to engage with their audiences and to maintain and even expand their popularity. Thus, social media have become a primary source of information providing

up-to-date knowledge on a wide variety of topics, from major events to the opening hours of stores or what books or songs a particular celebrity likes. Coincidentally, such people and organizations often have dedicated Wikipedia pages, and thus corresponding entries in knowledge bases (KB) related to Wikipedia, such as DBpedia (Lehmann et al., 2015), YAGO (Hoffart et al., 2013), or Wikidata (Erxleben et al., 2014). Data in social media and KBs present opposite characteristics. On the one hand, KBs provide high-quality structured information (e.g., YAGO has 95% accuracy, Hoffart et al. 2013) that is easily accessible, e.g., due the use of open formats (RDF) and online publishing as Linked Open Data (LOD), while data from social media accounts is often noisy, unstructured, and hidden behind restrictive APIs. On the other hand, social media provide up-to-date (real-time) information, while contents in KBs may lag behind from hours to months. For Wikipedia-related KBs, such lag comprises both the time for changing the page (hours to months, based on popularity, Fetahu et al. 2015) and, for automatically extracted KBs like DBpedia and YAGO, the time for that change to propagate in the KB (months to years). Such lag may prevent using these KBs in some application scenarios.

In light of these differences, we developed the SocialLink[1] project to bridge the KB and social media worlds by linking KB entities to their corresponding social media profiles. The core motivation to do so is to enable knowledge transfer between the two (see Figure 3.1). Although such alignments do exist for a few entities in DBpedia and Wikidata, they only cover a very small portion of all the suitable linking targets they contain. SocialLink's ability to produce reliable and comprehensive alignments allows significant increases in available coverage benefiting the downstream tasks using such links. We will provide relevant statistics on this matter along with many of the potential and existing use cases in the subsequent chapters.

In this chapter, firstly, we describe the overview of the SocialLink linking approach by presenting its three constituent phases. Secondly, we introduce graph-based features trained from the large amounts of unsupervised data available on both the social media and the KB sides, which required complete redesign of the final candidate selection phase. In third, we provide extensive evaluation of the algorithm both with basic features and the extended ones. Finally, we will provide additional discussion around some of the design choices, such as the usage of the neural networks, the choice of textual embeddings and the choice of abstention mechanism.

**Linking approach** The SocialLink's linking approach (first introduced in Nechaev et al. 2017b and Nechaev et al. 2018b) is a three-phase pipeline that (i) gathers, indexes and stores the data necessary to perform the linking (data acquisition phase), (ii) proposes a set of candidate social media profiles for each entity in a KB (candidate acquisition phase)

[1] https://w3id.org/sociallink

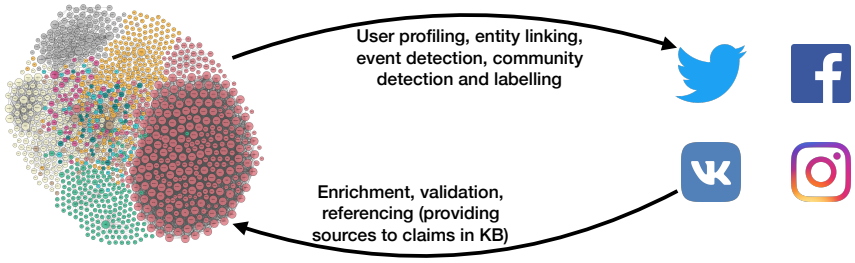



Figure 3.1. Knowledge transfer between the Linked Open Data cloud and social media

and (iii) uses a deep learning-based model to select (or abstain from selection) the best possible candidate for the target entity (candidate selection phase). Currently, SocialLink uses DBpedia as the KB and Twitter as the target social network. The approach is trained and evaluated using the 56,133 existing links to Twitter found in DBpedia and Wikidata, and is able to leverage large amounts of unsupervised data both from DBpedia and Twitter to improve performances. As the approach and most of the features it uses are general, it may be potentially expanded to support additional KBs and social media presenting characteristics similar to the ones considered here (e.g., availability of names, textual content and connections for both KB entities and social media profiles). The approach described here is used to build the complementary LOD-compliant dataset (will be covered in details in Chapter 4) that consists of more than 322K high quality (more than 90% precision) alignments and more than 1M candidate alignments, obtained by applying the above linking approach to 2M living people and 500K currently existing organizations in DBpedia (entities from multiple DBpedia language chapters are considered).

**Graph features** SocialLink is designed to exploit multiple feature types that can be derived from user and entity data. Many of the feature types, such as text and carefully selected metadata which we collectively refer to as BASE features, were successfully embedded into our neural network-based system without experiencing significant design complications. Embedding graph-based features into our system, on the other hand, has proven to be a hard challenge to overcome and constitute a significant portion of the novelty of this chapter from the algorithmic point of view.

*Graph-based features*, both on the social media and KB sides, is an essential source of information. On the social media side, the *social graph* plays a crucial role for many social media-related tasks, where it reveals a significant amount of information about the users. The social graph is basically a vibrant network of connections between users that is typically represented by an explicit "follow" action, which manifests the intent of a user (follower) to read content written by the followed user (friend). As will be shown in Chapter 6, using SocialLink along with a simple rule-based technique one can infer interests of a passive user (i.e., a user not generating any content on his own) by exploiting

just the user's social graph. The social graph has also been used to determine user's location (Sadilek et al., 2012), gender, and political affiliation (Zheleva and Getoor, 2009). Moving to the KB side, knowledge is often encoded in RDF triples and can naturally be represented as a knowledge graph with entities as vertices and relations as edges. These connections between entities are a powerful mechanism used in literature for solving a wide range of tasks both in the Semantic Web and many other domains (Ristoski and Paulheim, 2016b). Due to difficulties in acquiring and encoding such graph-based features, initial versions of SocialLink have utilized them only indirectly, through measures such as the number of friends and followers of a social media profile or the indegree and the outdegree of a KB entity.

To allow SocialLink to exploit such features explicitly, here we study the addition of graph-based features into the feature space in the form of *embeddings*, i.e., low-dimensional vector representations of nodes learned using large amounts of unsupervised data (see Section 2.5). Embeddings typically capture similarities among the objects they encode, a particularly useful trait for linking tasks like ours. On the social media side, we build on Swivel (Shazeer et al., 2016) to derive *social graph embeddings* for Twitter user profiles. In doing so, we address two fundamental challenges. Firstly, the social graph is typically very expensive to acquire at scale, as many social media obscure or hide the social graph altogether from third parties or, where it is available, significantly limit the number of user connections that can be sampled over a fixed period. In this chapter, we show that the social graph can be efficiently approximated using retweet and mention relations mined from the sampled tweet stream provided by the Twitter Streaming API (the same stream leveraged in the *candidate acquisition* phase to overcome similar limitations in the retrieval of candidates). Secondly, there are much more users in the social media than entities in the KB or words in any reasonable vocabulary, and most of the embedding generation approaches are not reasonably scalable to accommodate the complete social graph of any major social media. For example, Cochez et al. (2017b) provide embeddings with one of the largest vocabularies we have seen, yet it is still two orders of magnitude smaller than what is required by our task. We solve this issue by training embeddings only for the limited subset of the most followed users on Twitter and then treating other users as a weighted sum of the users they follow. Moving to the KB side, we leverage recent research results on RDF-based embeddings by Cochez et al. (2017b), using precomputed models made available by them (Ristoski et al., 2017) that cover almost 9M entities from the English DBpedia. In particular, we have found that among the models based on GloVe (Pennington et al., 2014), the PageRank-based weighting schemes provide the best improvement in our task, consistently with the general findings reported by the authors.

The introduction of the new features inevitably required modifications to our base model. Initially, we have tried to use the embeddings by directly concatenating them

with our old feature sets. However, since we are simultaneously adding vectors from two completely independent feature spaces, it becomes hard for the densely connected neural network to consistently find a solution that brings an improvement to the whole task. To this end, we show that by changing the topology of the network with the addition of a transformation layer followed by the multiplication of the transformed embeddings, instead of their simple concatenation, we are able to assist the training algorithm in finding a better solution overall.

**Evaluation** We provide the extensive evaluation of all the presented approaches. This evaluation is performed on the latest version of our gold standard for English DBpedia entities, which includes recent fixes and additions provided by the DBpedia and Wikidata communities, resulting in an overall growth from 35,149 we used during our initial experiments on this task (Nechaev et al., 2017b) to 56,133 alignments. We evaluate the two model variants presented (with or without employing the graph-based features) and the two baselines on this gold standard, showing that a trivial solution can not exhibit acceptable performances for this task and that our most complete approach offers significantly increased performance levels. In addition to raw numbers, we conduct the complete analysis of the errors this model makes to provide the more accurate assessment of the approach capabilities.

The rest of the chapter is structured as follows. In Section 3.2 we introduce the linking task in more details. Section 3.3 provides an overview of the SocialLink pipeline including the description of features in the base model. Section 3.4 introduces the new graph-based features, while Section 3.5 describes the neural model exploiting them. The evaluation of the new additions is provided in Section 3.6. In Section 3.7 we provide additional experiments to validate the choice of word embedding algorithm. We present related work in Section 3.8 and conclude summarizing our findings in Section 3.9. Source code and documentation for systems and approaches described in this chapter are available in our GitHub repository.[2]

## 3.2 Problem Definition

Our goal is to find a profile of an entity (person or organization) in a particular social network given the knowledge base (KB) entry for the entity, which consists of a set of attributes about the entity.[3] In the following, we consider the DBpedia KB and the Twitter social network, although the task definition and most of the remarks here are general and may apply to other KBs and social media with similar characteristics.

[2] https://github.com/Remper/sociallink

[3] We start from KB entries as they are entirely known in advance, differently from social network profiles that can be only queried or (partially) acquired via expensive crawling.

The information available in a KB entry depends on the KB considered and, within the same KB, may be different from entity to entity. DBpedia itself, although based on Wikipedia that is being updated by millions of people every day, can have various issues including inconsistency, noisiness, obsolete knowledge and unavailability of entity attributes. An entry about the President of the United States can, for example, contain many attributes from different domains, while an entry for a regional-level politician in a non-English speaking country can basically contain a name, a description, and the occupational class. This heterogeneity requires an approach that can work with the bare minimum of information known about the target entity. Here, we assume that a KB entry at least contains the name and a textual description, person vs organization type information, and some temporal information allowing the distinction between living/existing entities and non-existing ones. While we do not require additional information (types and relations), if present, such information can be exploited as graph features.

Working with social media from the outside also imposes a number of challenges. First, similar to KB entities, also the amount and quality of information available in a social media profile may vary. Specifically, profiles can be private, have limited attributes available, and / or contain confusing or inaccurate information. Therefore, a linking approach has to use the attributes that are most widely available in social media. They include, for example, user name, social graph, posting behavior, textual description and user-generated content and a special "verified" flag issued by the social media that certifies the identity of the profile owner.

Second, for famous people and organizations, there typically exist impersonating and fan profiles that can be very similar to the real one. Since most of the entities do not try to acquire the "verified" flag, it can be hard even for a human to distinguish them. Moreover, certain groups of people, e.g., politicians and athletes, tend to have multiple profiles that correspond to various periods in their life. A politician might create a new profile if he was elected, an athlete can do the same when changing teams. Some famous people tend to have an official and a personal profile. In all these cases, finding the right profile among very similar options is hard.

Third, for Twitter and many other social media it is not feasible to acquire the entire social network, due to its enormous size and the API limitations. Therefore, to acquire the candidate profiles for a target KB entity one has to use the available API request quota sparingly, which limits the amount of candidates and the types of candidate information that can be acquired. While these limits can be softened using more sophisticated API crawling techniques (in accordance with terms of use) and by leveraging possible information streams provided by the social network, like the sampled tweet stream we exploited in our work, one cannot generally assume that all relevant candidates and their information can be accessible when linking an entity, if not only for the difficulty of

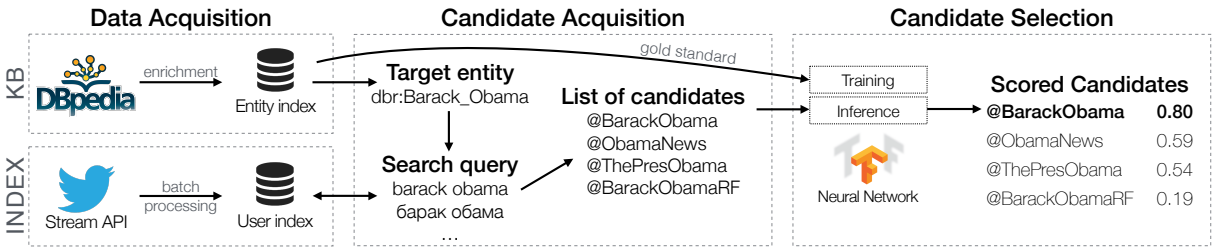



Figure 3.2. The three processing phases of the SocialLink pipeline.

processing such huge amount of information (e.g., running a classifier over millions of users for millions of KB entities). Therefore, in all the cases where it is impossible to link an entity to a profile we cannot infer that the entity is not present on the considered social network, as the right profile for the entity may just be not accessible and thus unknown (open world assumption).

The task that we solve here is similar to the well-known problem of profile matching on social media, if we look at a KB entry as a special kind of profile. However, KBs do not contain attributes that were vital to matching profiles in previous studies, such as usernames, user-generated content, and social graph. Therefore, the techniques outlined in such studies cannot be directly applied in our case and cannot provide a baseline for evaluating our approach.

## 3.3 Approach Overview

In this section, we provide an overview of SocialLink approach for linking DBpedia entities to Twitter user profiles. We describe the three-phase pipeline of SocialLink as well as detail the BASE feature set. The complete model, including the graph-based features, will be further detailed in Section 3.4 (embeddings) and Section 3.5 (improved selection model).

Figure 3.2 highlights the three phases of the approach. Processing starts with the *data acquisition* phase (Section 3.3.1), where the required data from Twitter and DBpedia, enriched with fresh data from Wikidata and including preexisting gold standard alignments, are gathered, prepared, and indexed locally for further processing. Next, in the *candidate acquisition* phase (Section 3.3.2), for each DBpedia entity a list of candidate Twitter profiles matching the entity is obtained by querying the indexes. Finally, the *candidate selection* phase (Section 3.3.3) uses the gold standard to train a neural network that scores and selects the best matching candidate, or abstains if there is no suitable candidate.

### 3.3.1 Data Acquisition

During this phase, we gather and process large amounts of data to support further steps. This includes the retrieval and local indexing of entity data (RDF triples) from DBpedia

Table 3.1. Information in gold standard and DBpedia 2016-04, including average number of names and description length (chars)

| | Live Entities (per, org) | Persons percentage | Average number of names | Average description length |
|---|---|---|---|---|
| Gold standard | 56,133 | 72.98% | 1.81 | 547 |
| DBpedia (EN) | 1,123,735 | 71.05% | 1.89 | 525 |
| DBpedia (All) | 2,589,023 | 78.62% | 1.61 | 518 |

and Wikidata and of user profile data from Twitter, as well as the generation of entity and profile *embeddings*, i.e., dense vector representations of objects learned from large amounts of unlabeled data and used as features in our approach.

**Entity Indexing** Entity information exploited in SocialLink consists of names, types, textual descriptions, live / not-alive status, relations to other KB entities, and preexisting alignments to social media profiles that form our gold standard. This data is mainly acquired from DBpedia[4] for *live* person and organization entities,[5] which account for the majority of the available DBpedia-Twitter alignments. We further enrich this data with more up-to-date alignments and entity death / closing dates from Wikidata, mapping from Wikidata entity identifiers to DBpedia entity identifiers using the owl:sameAs links from DBpedia and the RDFpro (Corcoglioniti et al., 2015) tool for URI rewriting. To speed up processing, and overcome the limitations of public SPARQL endpoints, we build a local *entity index* consisting of a Virtuoso triplestore populated with all the required data. Here the 56,133 gold standard alignments (40,967 persons, 15,166 organizations, available on our website) are also extracted to be used for training the candidate selection phase.[6] Table 3.1 provides relevant statistics for the gold standard, compared to DBpedia in general (English chapter and all chapters, linkable live entities only).

**Twitter Profile Indexing** Profile information exploited in SocialLink consists of names, user-generated texts, and social relations (e.g., follow, retweet, reply and mention relations). This information can be obtained from the social media API, that for Twitter consists of either the ReST API or the Streaming API. We employed the first initially (Nechaev et al., 2017b), but its rate limits (e.g., 180 user queries every 15 minutes) make it difficult

[4]English DBpedia version 2016-04, for what concerns the experiments reported here (to enable comparison of different alignment techniques developed in this thesis at different points of time).

[5]Entity alive status is gathered from temporal properties like `dbo:deathDate`, `dbo:deathYear`, `dbo:closingYear`, `dbo:closed`, `dbo:extinctionYear`, `dbo:extinctionDate`, `wikidata:P570`, `wikidata:P20`, `wikidata:P509`, or properties implying death like `dbo:deathPlace`, `dbo:deathCause`, `dbo:causeOfDeath`.

[6]Gold alignments derive from selected foaf:isPrimaryTopicOf and wikidata:P2002 triples of entities assumed living.

to acquire data for all the *candidate* Twitter profiles that may be linked to DBpedia. Consequently, we switched to the Streaming API (Nechaev et al., 2017d) that provides a continuous (sampled) stream of tweets posted on Twitter, each one enriched with metadata and information about its author.

From the Twitter stream, we continuously collect and index the following data. First, the latest profile encountered for each user is recorded. Since the Streaming API is allowing us to observe only a subset of the complete Twitter stream, it is not guaranteed that all possible profiles will be extracted. However, this approach has yielded 291M most active users, which we consider sufficient for our task. Secondly, all the available text generated by each profile is gathered. This includes the tweets written by a user as well as the (time-varying) textual descriptions found in the user profile for the observed time period. Third, we extract the names related to a profile along with the frequency for each name. Finally, we record the interactions between users by extracting mentions and retweets. This information is needed to acquire an approximate social graph for each profile.

Collected user data is indexed in a PostgreSQL relational database and a basic full-text search index is built to allow efficient searching of profiles by name. Data processing is implemented using Apache Flink,[7] a framework providing reliability (via automatic checkpoints) and scalability (via automatic horizontal scaling). We have currently gathered more than four years of raw Twitter data, out of which 450 GB of indexed and accessible user data are produced. This setup of the system allows performing hundreds of queries per second on a single machine and enables frequent, reliable, and fully automatic population and update of the SocialLink dataset. Additionally, the user index provides much more user-related data compared to live querying of Twitter ReST API, thus increasing alignment performances in both candidate selection and candidate acquisition phases.

**Embeddings** To effectively exploit textual information of DBpedia entities (short abstracts) and Twitter profiles (user descriptions, tweets) and deal with lexical variability, in our work (Nechaev et al., 2017b,d) we leverage low-dimensional vector representations of texts — i.e., embeddings — computed using a Latent Semantic Analysis (LSA) approach (Landauer et al., 1998; Cristianini et al., 2002). These word embeddings are derived from the term-by-document matrix of the English Wikipedia via dimensionality reduction (matrix factorization via singular value decomposition).[8] It is worth noting that a number of approaches were proposed to improve word embeddings in various ways over the last five years. We have conducted additional tests (see Section 3.7) with some of them and, since such approaches did not introduce significant improvement on our task,

[7] http://flink.apache.org/

[8] See (Aprosio et al., 2013, Section 3.2) for a detailed description of how LSA embeddings are computed.

we decided to stay with LSA to perform a proper comparison with the model we initially developed as part of this thesis.

In addition to LSA embeddings, which account for text information only, we introduce *graph embeddings* to account also for relational data, both in the KB and the social media. On the KB side, we use the 'PageRank Split' variant of graph embeddings described in Cochez et al. (2017b) and Cochez et al. (2017a) and computed for the English DBpedia (version 2016-04),[9] which are a particular implementation of *RDF graph embeddings* (Ristoski et al., 2017). On the social media side, we propose our own *social graph embeddings* that capture the information of a user's social relations, as detailed in Section 3.4. Together, these graph embeddings allow exploiting the large amounts of unlabeled RDF and Twitter data available online and in our indexes, to learn effective low-dimensional representations for entities and user profiles that can be exploited in the alignment task.

### 3.3.2 Candidate Acquisition

In this phase, given an entity to align to Twitter, we obtain a list of candidate Twitter profiles that is expected to contain the true candidate for the entity, if any. During our initial experiments (Nechaev et al., 2017b) Twitter ReST API was used to produce the candidate list. Currently, we employ a full-text search query targeted at our user index (Nechaev et al., 2017d). In both cases, the choice of query is significant: we want to maximize recall, i.e., the probability of finding the true candidate among the list, without introducing too much noise (i.e., unrelated candidates) that may decrease performances in the following selection phase.

We have thoroughly investigated different query construction strategies (detailed in Nechaev et al. 2017b) and settled for the strategy that combines all known names of a DBpedia entity as encoded by properties foaf:name and rdfs:label. Names consisting only of first or last name (i.e., foaf:name matching foaf:givenName or foaf:surname) are filtered out to prevent noisy results. Indeed, if for entity "John Smith" we were to keep the name "John", the list of potential candidates would contain all possible Johns, which is not desirable. The remaining names are deduplicated and the three most frequent ones (in collected foaf:name and rdfs:label properties) are OR-ed to form the query, as shown in the example in Table 3.2. If no results or too many results are obtained, the query is modified by selecting a different combination of names. The returned results are sorted based on the frequency with which we saw a name referring to a candidate profile in a stream of tweets. In the end, we save at most the top $k = 40$ results returned by the user index as

[9] PageRank Split embeddings downloaded from http://data.dws.informatik.uni-mannheim.de/rdf2vec/models/DBpedia/2016-04/GlobalVectors/

Table 3.2. Example of candidate retrieval query.

| | |
|---|---|
| Entity | `http://dbpedia.org/resource/Barack_Obama` |
| Names | `Barack Obama`, `Barack Hussein Obama` |
| Query | `(Barack Obama) OR (Barack Hussein Obama)` |
| Result | `@BarackObama` *(true candidate)*, `@ObamaNews` ... *other 38 candidates* |

candidates for the entity, this threshold empirically chosen with the goal of maximizing recall while reducing noise. So if the candidate with a name-based match is not mentioned or retweeted as much as other matching candidates, it would not appear in the resulting list.

Compared to live querying the Twitter ReST API, the chosen approach based on the user index allows a greater degree of flexibility in acquiring the candidate list for a DBpedia entity, as it enables different query strategies and allows bypassing API limitations in terms of query complexity and request rates (at most one request / 20 candidates per entity could be feasibly obtained with the ReST API), resulting in an increase of recall (see comparison between $v1.0$ and $v3.0$ in Chapter 4).

### 3.3.3 Candidate Selection

In this phase, given a DBpedia entity and the corresponding list of candidate Twitter profiles, we formulate a classification problem where the classifier has to provide a probability estimate of a candidate being a match of the target entity, for each considered ⟨candidate, entity⟩ pair.

As classifier we employ a deep neural network (DNN)[10] trained on the gold standard DBpedia-Twitter alignments described earlier. Our initial DNN model (Nechaev et al., 2017b,d) consisted of a stack (5 hidden layers with 256 units each) of densely-connected layers with *tanh* as activation function and *softmax* applied on top to acquire probability estimates. The DNN takes a feature vector as input consisting of the features of Table 3.3 and all their pairwise combinations, scaled to unit variance and zero mean and hereafter referred to as BASE features. The *Adam* algorithm is used to train the network, employing cross-entropy as cost function. Dropout with the probability of 0.5 is applied to each layer, and L2 regularization is used to prevent overfitting. In the next two sections, we revise and improve this architecture to include graph embeddings both for the entity and the candidate profile, as described in Section 3.5.

After the probability estimates are computed, the candidate with the best probability is selected. SocialLink can abstain from selection based on a minimal score threshold.

[10]For the BASE feature set simpler models, such as SVM-based ones, were tested during our initial experiments on this task, please refer to (Nechaev et al., 2017b) for more details.

Table 3.3. Basic features used in the DNN classifier. In Nechaev et al. (2017b) we have also considered Wikipedia-specific features that are not covered here. As their contribution to the system performance was limited, we do not employ such features here since they will make it harder to adapt SocialLink to target non-Wikipedia-based KBs.

| Feature family | Feature type | Feature description |
|---|---|---|
| Name | 4 scalars | edit distances are computed for pairs ⟨entity name, profile username⟩ and ⟨entity name, profile screen name⟩ using two metrics: Jaro-Winkler and Levenshtein resulting in four scores. Since an entity can have an arbitrary amount of names we average the scores across all of them. Such features are known to be useful when aligning profiles (Zafarani and Liu, 2013, 2009). |
| Description | 2 scalars | two cosine similarity scores between the entity description (rdfs:comment) and the two texts derived from user-related textual content in Twitter: profile text (description, location, pinned tweet) and the content of tweets made by the user (as extracted from our stream of tweets). The average of tf-idf-weighted word embeddings was used to represent text. |
| Core profile metrics | 4 scalars | logarithms of friends, followers, tweets and listed counts for the profile, to measure the popularity and activity level of the Twitter profile. These features capture the intuition that the true candidate profile for a KB entity is often the most popular and/or active one (this is especially the case for famous KB entities). |
| | 1 binary | set to 1 if the profile has been 'verified' by the social media provider. Even if the percentage of verified entities is rather low, it is still one of the most effective features to help distinguishing a real profile from a fake one. |
| Homepage links | 3 binary | we crawl links to Twitter profiles in DBpedia entity homepages (property foaf:homepage) and define three features to specify: (i) if the profile is contained in the list of profiles scraped for the entity; (ii) if the profile is the only one extracted; and (iii) if the profile contains a link back to the crawled homepage. |
| Entity type | 2 binary | entity type chosen among person (dbo:Person type) and organization (dbo:Organization type), to permit the DNN classifier to learn a different strategy for persons and organizations. |

Section 3.6 provides precision / recall curves produced by changing this threshold during the evaluation on the gold standard dataset.

## 3.4 Graph-based Embeddings

The addition of graph-based embeddings in our case enables exploiting connections among entities and users. Embeddings provide a dense low-dimensional representation of objects trained to reflect similarities between them. In case of neural language models, for example, individual embeddings typically follow the *distributional semantics* hypothesis: words that are used in the same contexts tend to have similar meanings, which means that in the resulting vector space such words will be close to each other according to some distance metric. Inspired by the overwhelming success of word embeddings in many tasks, similar approaches appeared for representing nodes in graphs. They follow the same principle: nodes that have similar neighborhood in a graph will have similar representations.

In this section, we motivate and describe the algorithm we have chosen to represent graph-based features from DBpedia, as well as present our novel approach for the social graph in Twitter. Both algorithms are based on neural language models, specifically, the global log-bilinear regression models that produce embeddings by factorizing[11] the co-occurrence matrix of objects (entities in case of DBpedia, users in case of Twitter). We refer the reader to Chapter 2 for additional background information on graph embeddings and the state-of-the-art algorithms we leverage and extend here.

As mentioned in Chapter 2, in order to construct the aforementioned co-occurrence matrix following the distributional semantics hypothesis, one has to define the focus object and one or many context objects. In case of natural language, where the objects are words, for each focus word in a sentence, some neighboring words are selected as its context. Each such occurrence is weighted based on a weighting function, such as $1/d$ if the two words are $d$ words apart, and added to the co-occurrence matrix.

Following this approach through an unsupervised corpus results in the co-occurrence matrix $\mathbf{X} \in \mathbb{R}^{v_f \times v_c}$, where $v_f$ is the size of the vocabulary of focus objects and $v_c$ is the size of the vocabulary of context objects (for simplicity, the same vocabulary size is typically used for both). Then, $\mathbf{X}$ is factorized into matrices $\mathbf{W} \in \mathbb{R}^{v_f \times d}$ and $\tilde{\mathbf{W}} \in \mathbb{R}^{v_c \times d}$, where $d$ is the desired embedding size, by minimizing the objective from Eq. 2.3. The model with this objective is referred to as the GloVe model Pennington et al. (2014) and effectively encourages $\mathbf{W}\tilde{\mathbf{W}}^\top$ to predict $\log \mathbf{X}$. Swivel, instead estimates the pointwise mutual information between the terms ($\text{pmi}(i; j)$), modifying the objective (see Eq. 2.5 and 2.6).

[11] The regression models described in this section perform an approximate matrix factorization rather than the exact one used, for example, by LSA.

Given that the notion of focus and the context objects can be arbitrarily defined, those approaches can be applied to any task including the representation learning of nodes in a graph, provided a sufficiently large unsupervised dataset from which to extract the co-occurrence statistics is available (as in our case).

### 3.4.1 Social Graph Embeddings

Our user index (described in Section 3.3.1) contains 291M Twitter users during the *candidate acquisition* step and this number continues to grow as we sample more data from Twitter. Each of those users can potentially end up being a candidate and would require a full set of features at the *candidate selection* step. Similarly to RDF graph embeddings, we would like to have vector representations for users in the social media based on the social graph capturing similarities between those users. Here we present an approach for computing such embeddings having in mind two fundamental issues that inevitably arise when trying to learn representations based on the social graph: the acquisition of the social graph and the expected coverage.

Social media APIs are incredibly restrictive: at the time of writing, Twitter allows only one request per minute covering at most 5,000 social graph edges at a time. To resolve this issue, instead of acquiring the exact social graph for each user, we record interactions, such as retweets and mentions, between users extracted from the Twitter Streaming API. In total, over 2.7B of such interactions were gathered. In Twitter, the users are unlikely to see a piece of content or a user that is not close to them in their social graph. This observation allows us to estimate the social graph by creating a "follow" edge for each such interaction. Additionally, we weight each edge in the graph by observing the locally normalized frequency of interaction: $\text{sg}_{ij} = \text{freq}_{ij} / \sum_k \text{freq}_{ik}$. So if the user is interacting with a particular set of profiles more often, such profiles will have a more prominent weight.

The growing number of Twitter profiles in our user index poses unique requirements when designing user embeddings. Firstly, the learning approach has to be scalable. Learning embeddings for each of the 291M users would be simply impractical. Even though many of the methods would scale linearly with the size of the vocabulary, when increased by two orders of magnitude learning time and memory requirements would still become unreasonably high. Indeed, most of the approaches would store embeddings in memory during training, which would quickly become a problem, especially if trained on a GPU. DeepWalk and node2vec use random walks, which are generated based on the number of nodes in the graph. The log-bilinear factorization approaches described above, in general, depend quadratically on the vocabulary size. However, GloVe in practice scales much better, since it is not utilizing unobserved co-occurrences in any way and the

co-occurrence matrix is typically increasingly sparse. By comparison, Swivel would not benefit from such sparseness.

Scalability issues aside, the approach has to be able to reliably represent out-of-vocabulary users, which would inevitably appear as we continue to sample the stream of tweets. To address both issues, instead of learning embeddings for each user, we learn them only for a subset of profiles that are likely to be followed (friends) by other users, and then represent any given user in Twitter as a weighted sum of their known friends. This procedure allows us to build embeddings for any given user as long as he is following some profile for which we have the embedding.

To this end, for each user in our estimated social graph, the list of friends was retrieved and represented as a sequence. Then, each profile in this sequence was treated as a focus object while the rest were treated as co-occurring context objects with unit weights. This mimics natural language models by essentially treating users as documents and friends as words where the particular order of words does not matter. Since the computation of such co-occurrences is quadratic in the size of the sequence and we do not get much information by observing users with a large number of friends, we limit the maximum sequence size to 30,000 to speed up the computation. This procedure yields a dataset of almost 200M sequences from which the 500K most frequent users were chosen to form a vocabulary, and the co-occurrence matrix was calculated. Note that we consider each co-occurrence equally important at this stage.

The co-occurrence matrix is then factorized using Swivel. Swivel objective (2.5), compared to GloVe, bears two characteristics essential for our task. First, training to estimate the pointwise mutual information instead of the raw co-occurrence gives us a natural way to punish profiles that are very frequent and, therefore, their occurring in the list of friends do not bear much information. Second, co-occurrence matrices of such size tend to become increasingly sparse. Considering that our estimation of the social graph is based on an incomplete sample of Twitter, Swivel's additional term (2.6) for the unobserved co-occurrences is particularly important.

After the factorization, the matrix of embeddings $U \in \mathbb{R}^{v_{\text{sg}} \times d_{\text{sg}}}$ including $v_{\text{sg}} = 499,712$ vectors with $d_{\text{sg}} = 300$ is produced. For the other users, an embedding is calculated as a weighted average of the embeddings of their friends:

$$\text{emb}_{\text{sg}}(\text{user}) = \begin{cases} \frac{\sum_{f \in \text{fr(user)}} U_f * \text{sg}_{uf}}{\sum_{f \in \text{fr(user)}} \text{sg}_{uf}}, & \text{if fr(user)} \neq \varnothing \\ \overline{u}, & \text{otherwise} \end{cases} \tag{3.1}$$

where fr(user) is a list of friends for which the embedding exists and $\text{sg}_{uf}$ is the weight for a social graph edge from $u$ to $f$. The mean embedding $\overline{u_j} = \sum_i u_{ij}$ is used if a target user does not follow anyone we have the embedding for.

### 3.4.2 RDF Graph Embeddings

KBs typically consist of RDF triples, providing a natural way of interpreting entities and relations between them as a graph. The goal of RDF-based embeddings is, therefore, to provide vector representations for each entity in the KB for computing similarities between entities in our task.

Cochez et al. (2017b) propose a number of approaches for building a co-occurrence matrix based on RDF data. They treat each node in the RDF graph as co-occurring with its neighbors and investigate the usage of Personalized PageRank (PPR) to determine the weights of such co-occurrences. They develop an efficient approximation of PPR based on the Bookmark-Coloring Algorithm and devise 12 weighting schemes to use along with it. After each edge in a graph is weighted, the co-occurrence matrix can be formed. Then the GloVe model is executed following the objective (2.3) and computing an embedding vector for each entity in the vocabulary (i.e., the RDF KB).

In our experiments, we tested the impact of those weighting schemes on our task using the precomputed embeddings provided by the authors, and we selected the "PageRank Split" variant that provides optimal performances. We confirm the authors' findings that the weighting scheme significantly affects the performances on a given prediction problem. The provided embeddings $K \in \mathbb{R}^{v_{\text{kb}} \times d_{\text{kb}}}$ include $v_{\text{kb}} = 8,876,676$ vectors with $d_{\text{kb}} = 200$ and cover most of the entities required by our linking task. In cases where the embedding of an entity is not available, the mean embedding $\overline{k_j} = \sum_i k_{ij}$ is used.

## 3.5 The Embedding-Aware Candidate Selection Model

Once embeddings from both the social graph and the RDF graph are available, we inject them into our *candidate selection* model introduced in Section 3.3.3.

The first aspect to consider concerns the way those embeddings are added to the model. The most trivial way of adding new features into the model is to just concatenate the old feature vector (BASE features) with both the embeddings for the candidate and the entity. However, considering that the BASE feature space contains just 136 dimensions, concatenating it with an additional 500-dimensional vector ($d_{\text{sg}} + d_{\text{kb}}$) would make it harder for the model training to arrive at a stable solution. Moreover, the BASE feature space contains features that were handcrafted for this specific task and can produce good performances on their own. We thus aim to add new features in a way that does not interfere with the network ability to learn from the BASE features too.

The second aspect to consider is that the candidate selection phase is a binary classification task, where the goal is to learn to match a particular entity with a particular candidate. So, the core motivation to have the graph-based features in the model in the

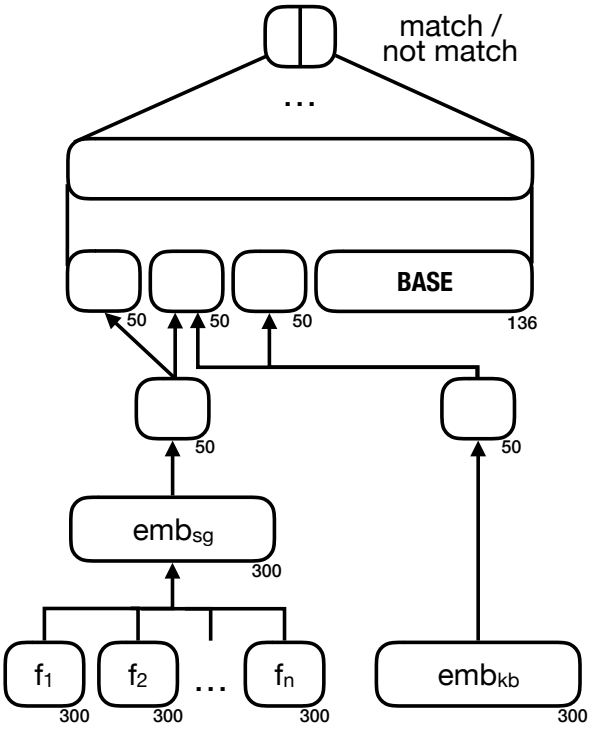



Figure 3.3. Schematic view of the updated candidate selection model, showing the new neural network architecture. The updated model is able to accommodate both the BASE features and the graph-based features as input through a special transformation layer.

first place is to exploit similarities between the RDF graph and the social graph. However, since both vector spaces of embeddings were trained separately, the model itself has to learn how to combine them in a useful way, which makes the optimization problem unnecessarily harder. During our early experiments with the simple concatenation approach, we were unable to produce a consistent enough improvement over the BASE model.

The third and last aspect to consider is that representation learning from unsupervised data is typically used as a pre-training step. Then it is customary for a neural network to modify those representations during back propagation to trade the initial generality for performance in a particular prediction task. Unfortunately, we cannot use the same technique here. We have built our approach to be general enough to produce a SocialLink resource targeting at least 2.5M entities, while our gold standard dataset contains only 56,133. It is unlikely that training on such a gold standard will modify over 9M embeddings in a way that would produce a good general solution for the larger task.

To address those three issues, we started with adding a densely connected layer after each of the embeddings. This *transformation layer* further reduces the dimensionality of embeddings to just $d_{\text{emb}} = 50$, acting as a global modifier of the entire embedding feature space instead of modifying individual vectors. To encourage the network to use this transformation layer to map the original embeddings into the same feature space,

we introduce component-wise combinations of features for the embeddings by adding a *multiplication term*. Then the multiplication term and the transformed embeddings are concatenated with the BASE features, producing a total of 286 features ($d_{\text{emb}} * 3 + 136$), which go through the same densely connected layers as before. By modifying the topology of the network in this way we were able not only to improve consistency during training but also to further improve the results. Parameter $d_{\text{emb}}$ was initially chosen so that embeddings terms together would be roughly the same size as BASE feature set. However, we did test larger values with an increment of 25 up to $d_{\text{emb}} = 150$. We did not notice significant performance changes with values above 100 requiring more epochs to converge. The rest of the network follows our previous approach described in Section 3.3.3. Figure 3.3 shows the updated network architecture.

Finally, to better exploit the classification outcomes produced by the revised model, we have also changed the way we interpret the probability estimates we receive from this pairwise classification model. After acquiring each probability estimate $p_i$ for candidates $i = 1 \dots n$, we have to decide whether to align the entity to a candidate $i$ or to abstain, i.e., decide that no candidate matches the entity, a case that we denote with *nil* (analogously to *nil* in Entity Linking literature). Lacking a specific estimate for $p_{nil}$, we empirically set $p_{nil} = 1 - \max_i p_i$ and then re-scale $p_i$, $i = 1 \dots n$, so that after normalization we have a proper probability distribution satisfying $p_{nil} + \sum_i p_i = 1$ (note that $\sum_i p_i$ may be originally greater than 1 as estimates $p_i$ are produced independently). We then select the candidate $i \in \{1 \dots n\}$ with the largest probability estimate, if greater than a probability threshold chosen to control the precision / recall balance. This estimate is released along the SocialLink dataset and provides an indication of the alignment reliability, permitting users to set a probability threshold to select only the most reliable alignments and thus operate different precision / recall balances.

## 3.6 Evaluation

We provide here an experimental evaluation of the linking approaches described in this chapter, starting by introducing the experimental setting (Section 3.6.1) and then reporting on the system performances on the overall linking task (Section 3.6.2) and, separately, on the two main phases this task is organized in: candidate acquisition (Section 3.6.3) and candidate selection (Section 3.6.4). We also analyze the performances on different entity and profile types (Section 3.6.5) and conduct an error analysis (Section 3.6.6).

### 3.6.1 Experimental Setting

We denote our most complete approach that is using all the available features as BASE_KB-_SG_TL, as it integrates base features (BASE) with knowledge base (KB) and social graph (SG) embeddings, combined using a translation layer (TL); we consequently concatenate different subsets of those feature identifiers to denote system variants obtained by ablation of selected features.

Since there are no publicly available datasets for our task, we evaluate BASE_KB_SG_TL and its variants on the gold standard collected from DBpedia and Wikidata, consisting of 56,133 alignments from English DBpedia entities (40,967 persons, 15,166 organizations) to corresponding Twitter profiles. Of these profiles, 94.69% are in our user index built from the tweet stream and may be successfully aligned by the system. We use stratified 5-fold cross validation, with alignments computed for each test partition being concatenated to form a single set of alignments for the whole gold standard, over which our analyses are performed.

As our main performance measures, we use *precision* ($P$), *recall* ($R$), and $F_1$ *score*. These measures are computed considering as true positive ($TP$) every alignment produced by the system that matches a gold standard alignment, as false positive ($FP$) every system alignment not present in the gold standard, and as false negative ($FN$) every gold standard alignment not found by the system, with $P = \frac{TP}{TP+FP}$, $R = \frac{TP}{TP+FN}$, and $F_1 = \frac{2 \cdot P \cdot R}{P+R}$. We report $P$, $R$, and $F_1$ scores together with their 95% confidence intervals computed via the *percentile bootstrap* method, and assess the statistical significance of the difference of those scores via the paired *approximate randomization* test (Noreen, 1989) (significant if $pvalue \leq 0.05$). Since the output of the system is a set of alignments each one associated to a probability estimate, different precision / recall balances and thus $P$, $R$, $F_1$ scores can be obtained by setting a varying threshold on that estimate. This gives rise to a precision / recall curve that we report and analyze to study the performance of the system in different settings that differ by the relative importance given to precision and recall. From that curve, we pick the threshold producing the best $F_1$ score, and use the corresponding $P$, $R$, $F_1$ triple as the overall assessment of the system performances.

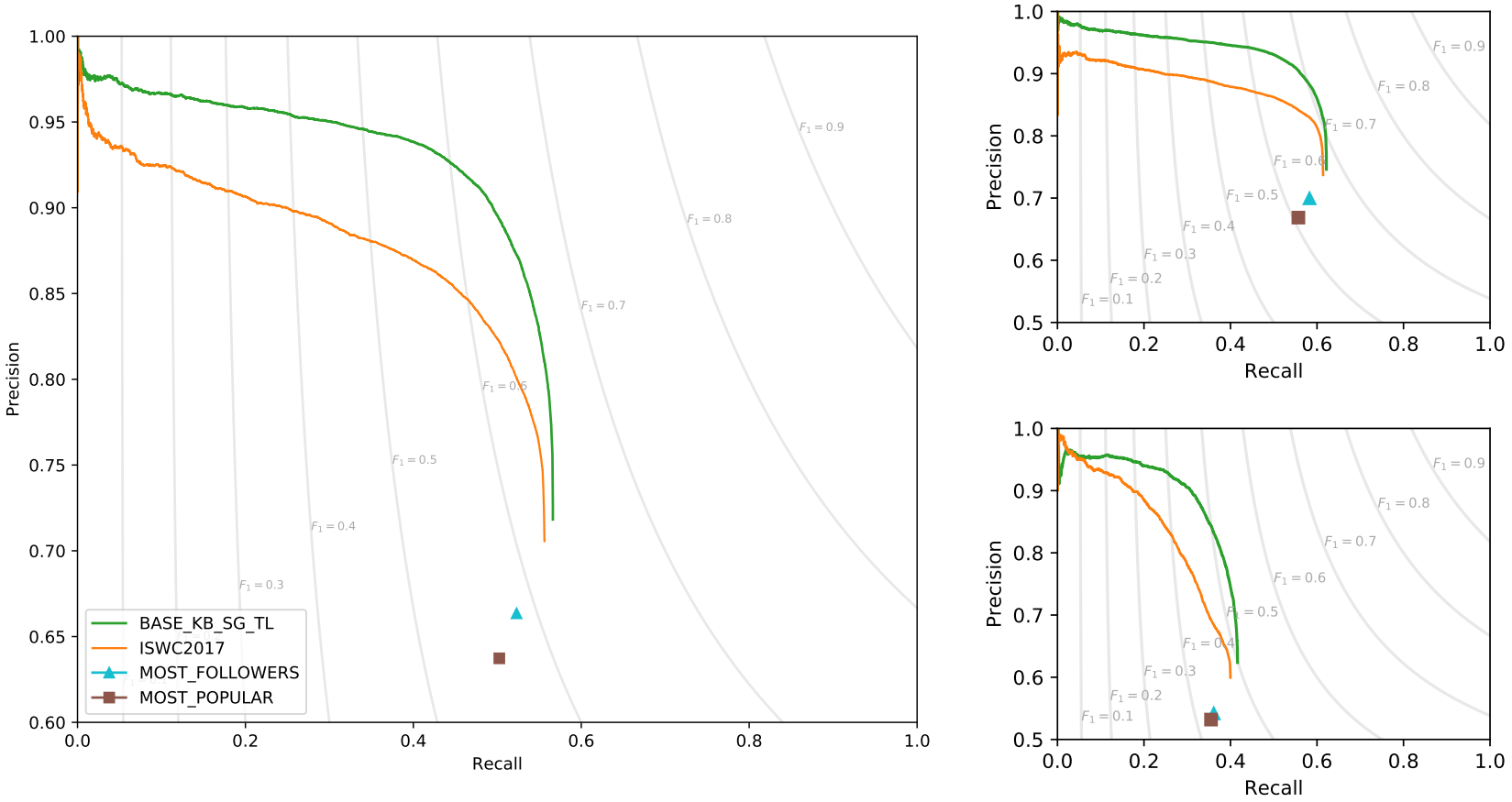
To assess the effectiveness of the complete system described in this chapter, we compare it against two baselines:

1. MOST_POPULAR. This baseline implements a straightforward approach where a DBpedia entity is aligned to the most popular Twitter profile matching it. Here, *matching* is defined as having one of the entity names in the profile full name or screen name, and thus being in the candidate list obtained by querying our user index. *Most popular* is defined as being the matching profile "seen the most" (i.e., authoring and/or being mentioned) in the tweets collected from the Twitter stream.

2. MOST_FOLLOWERS. Here we select as a correct alignment the Twitter profile with the most followers among the ones in the user index matched to the entity.

Additionally, we add a simpler version of the SocialLink pipeline, ISWC2017, that we previously utilized in (Nechaev et al., 2017b) and reused in (Nechaev et al., 2017d), which as described in Section 3.3.3 consists in a simpler DNN not leveraging relational information in the form of graph embeddings, and where the probabilities estimated by the DNN are directly used to select the best candidate without any normalization, differently from the new strategy that we described in Section 3.5. We include this model to showcase the cumulative effect of those additions to the SocialLink pipeline.

Note that the MOST_POPULAR baseline is very similar to the one we used in (Nechaev et al., 2017b) where we queried the Twitter ReST API (instead of the Streaming API) for a target entity name and we selected the first profile returned (if any) as the alignment for the entity. Both the old and the new baselines try to simulate a user's search for the matching profile, with the difference that we now search on our own user index built from Twitter data rather than querying Twitter directly, and we use the popularity notion defined above as a proxy for the ranking of query result provided by Twitter, which empirically appears to produce similar results. The MOST_FOLLOWERS baseline similarly captures the popularity of a candidate profile with an explicit metric: the audience size.




| Model | Persons | | | Organizations | | | All entities | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 |
| BASE_KB_SG_TL | 0.856 ±0.004 | 0.602 ±0.005 | 0.707 ±0.004 | 0.727 ±0.010 | 0.406 ±0.008 | 0.521 ±0.008 | 0.831 ±0.004 | 0.549 ±0.004 | 0.661 ±0.004 |
| ISWC2017 | 0.808 ±0.004* | 0.604 ±0.005 | 0.691 ±0.004* | 0.630 ±0.010* | 0.397 ±0.008* | 0.487 ±0.008* | 0.768 ±0.004* | 0.548 ±0.004 | 0.639 ±0.004* |
| MOST_FOLLOWERS | 0.700 ±0.005* | 0.583 ±0.005* | 0.636 ±0.005* | 0.542 ±0.010* | 0.361 ±0.008* | 0.434 ±0.008* | 0.664 ±0.004* | 0.523 ±0.004* | 0.585 ±0.004* |
| MOST_POPULAR | 0.669 ±0.005* | 0.557 ±0.005* | 0.608 ±0.005* | 0.532 ±0.010* | 0.355 ±0.008* | 0.426 ±0.008* | 0.637 ±0.004* | 0.502 ±0.004* | 0.562 ±0.004* |

Figure 3.4. P/R curves of overall system: (a) all entities; (b) persons; (c) organizations; (d) precision, recall, and F1 scores for the setting maximizing F1, with confidence intervals and statistical significance (*) of difference wrt. best model

Concerning the DNN-based ISWC2017 system, in (Nechaev et al., 2017b) we have also investigated the use of an SVM for the candidate selection phase, obtaining slightly worse $F_1$ scores when tuning the system for $F_1$, and worse $P$ scores when tuning the system for precision. We refer the reader to (Nechaev et al., 2017b) for further details.

### 3.6.2 Overall System Evaluation

Figure 3.4(a) reports the precision / recall curve of the improved BASE_KB_SG_TL system, compared on the same data to the curve of the ISWC2017 system and to the single $\langle P, R, F_1 \rangle$ points of the MOST_POPULAR and MOST_FOLLOWERS baselines. Figures 3.4(b) and 3.4(c) provide the same comparison restricted respectively to the alignment of person and organization entities only.

The BASE_KB_SG_TL system outperforms all other systems for all the $P$ / $R$ balances, except for low recall values for organizations. $P$ and $F_1$ differences for the system configurations maximizing $F_1$ are always statistically significant, as shown in Table 3.4(d), while $R$ differences are not statistically significant when compared to the ISWC2017 approach.

Both figures and table show that persons are aligned better than organizations. This difference is exhibited also by the considered baselines and is consistent with our initial results (Nechaev et al., 2017b), suggesting that the considered linking task is inherently more difficult for organization entities.

Both the BASE_KB_SG_TL and the other systems exhibit a sensible loss in terms of recall. Part of the loss is explained by the fact that our user index contains only 94.69% of the profiles in the gold standard, which bounds the recall of any approach using the index to 94.69% and thus limits the $F_1$ score to 97.27%. The additional loss of recall can be explained by separately analyzing the two phases of the task.

### 3.6.3 Candidate Acquisition Evaluation

We analyze the internal behavior of the system starting from the candidate acquisition phase. Table 3.4 provides relevant statistics for this phase computed for all gold standard entities and for persons and organizations only. They include: (i) the percentage of entities for which some candidate is returned by querying the user index; (ii) the percentage of entities for which the returned candidate list contains the true candidate for the entity, i.e., the candidate acquisition *recall*; (iii) the average length of the returned candidate list, when not empty; and (iv) the average position of the true candidate in the candidate list, when not empty.

Similarly to what observed in the overall evaluation, candidate acquisition for person entities exhibits better performances (recall, average true candidate index, non-empty

Table 3.4. Candidate acquisition statistics per entity type

| Entity type | Has some candidate | Has true candidate | Avg. candidates | True candidate avg. index |
|---|---|---|---|---|
| Persons | 83.3% | 65.1% | 9.8 | 1.7 |
| Organizations | 66.7% | 46.1% | 10.5 | 2.3 |
| All entities | 78.8% | 60.0% | 9.9 | 1.9 |

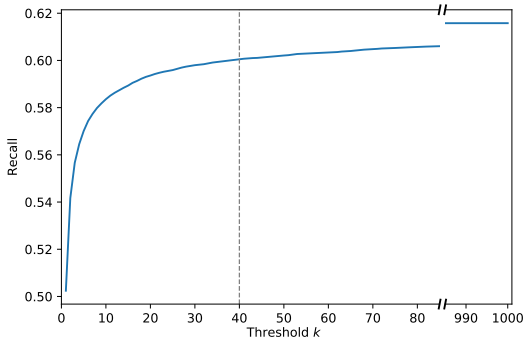



Figure 3.5. Candidate acquisition recall, i.e., fraction of entities whose fetched candidate list contains the true candidate

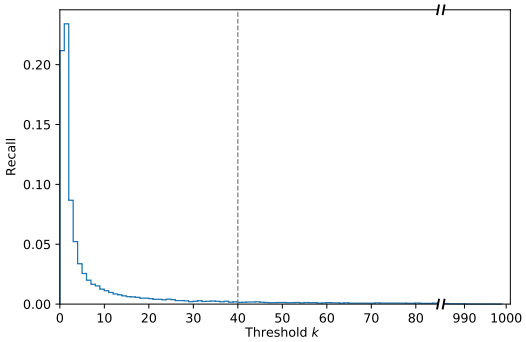



Figure 3.6. Frequency distribution of the number of candidates fetched per entity using our candidate acquisition strategy

candidate lists). For both kinds of entities and overall, the recall levels are limited, with the achieved recall of 60% (all entities) introducing a further loss of 34% to the 6% loss due to the right profile not being present in the user index. This additional loss reflects the difficulty of matching the entity names in the KB to the names used in the social media, which in some cases may be completely different. Since there is no way for the system to produce an alignment if the true candidate is not in the candidate list, the low recall levels obtained in this phase set hard limits to the recall achievable by the system on the overall task.

Table 3.4 is complemented by Figure 3.5 that shows the recall obtainable by varying the maximum number of candidates $k$ fetched for each entity (see Section 3.3.2), and Figure 3.6 that shows the frequency distribution of the number of candidates fetched for an entity; the vertical lines correspond to the chosen threshold $k = 40$. The histogram shows that very few entities in the long tail have more candidates than the threshold $k = 40$. This is reflected in the recall plot of Figure 3.5, where going from the selected $k = 40$ to an hypothetical $k = 1000$ will bring only a 1.57% increase in recall (from 60.01% to the "plateau" value 61.58%), at the price however of a sensible increase in computation costs and a possible loss of precision in the overall task due to the introduction of noisy candidates in the input to candidate selection. Therefore, $k = 40$ represents a good trade-off solution.

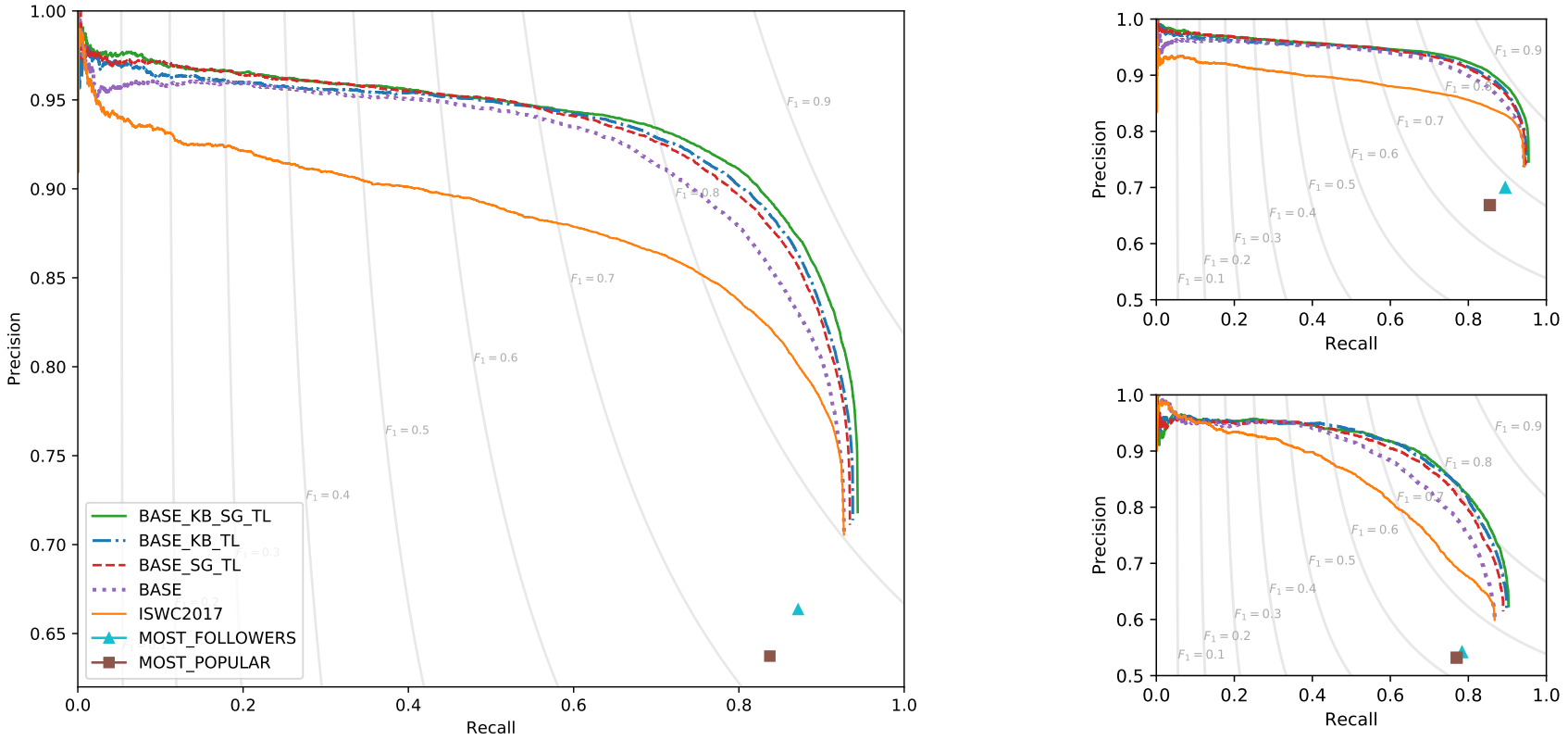
### 3.6.4 Candidate Selection Evaluation

We now turn our attention to the candidate selection phase, whose performances are assessed in terms of precision / recall for the only entities for which a non-empty set of candidates is returned by the candidate acquisition phase.

Figure 3.7(a) shows the precision / recall curves of BASE_KB_SG_TL, ISWC2017, the two baselines, and three variants of the proposed system: the BASE_KB_TL and BASE_SG_TL variants, obtained by removing the multiplication term from the DNN and, respectively, the social graph and the KB embeddings, and the BASE variant, obtained by removing both kinds of graph embeddings but keeping the normalization of probabilities and the other changes applied to the pipeline with respect to the ISWC2017 system. Figures 3.7(b) and 3.7(c) provide the same comparison restricted however to person and organization entities only, respectively. Table 3.7(d) reports $P$, $R$, and $F_1$ scores for the configurations with the best $F_1$.

As for the overall task, BASE_KB_SG_TL consistently outperforms the evaluated systems, with $P$, $R$, $F_1$ differences always statistically significant except overall recall with respect to the ISWC2017 system. Notably, recall is high in this phase, implying that the loss of recall in the overall task is largely due to the candidate acquisition phase. Also in this phase, performance is marginally better for persons.

When comparing the full BASE_KB_SG_TL system to its BASE_KB_TL, BASE_SG_TL, and BASE variants obtained via ablation of graph-based features, both figures and table show that each ablated feature contributes positively to the performances of the system, with the best performances achieved by combining both types of graph embedding features. In particular, Table 3.7(d) shows that the $F_1$ improvements for all entities obtained by adding features are always statistically significant. Finally, the positive impact of the new probability normalization and DNN fine tuning is demonstrated by the difference in performances between the BASE variant and the ISWC2017 system.




| Model | Persons | | | Organizations | | | All entities | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 |
| BASE_KB_SG_TL | 0.873 ±0.004 | 0.908 ±0.004 | 0.890 ±0.003 | 0.802 ±0.009 | 0.822 ±0.009 | 0.812 ±0.008 | 0.856 ±0.004 | 0.892 ±0.003 | 0.874 ±0.003 |
| BASE_KB_TL | 0.866 ±0.004* | 0.901 ±0.004* | 0.883 ±0.003* | 0.818 ±0.009* | 0.797 ±0.010* | 0.808 ±0.008 | 0.860 ±0.004* | 0.876 ±0.004* | 0.868 ±0.003* |
| BASE_SG_TL | 0.863 ±0.004* | 0.900 ±0.004* | 0.881 ±0.003* | 0.798 ±0.010 | 0.799 ±0.010* | 0.798 ±0.008* | 0.855 ±0.004 | 0.874 ±0.004* | 0.864 ±0.003* |
| BASE | 0.843 ±0.004* | 0.901 ±0.004* | 0.871 ±0.003* | 0.783 ±0.010* | 0.769 ±0.010* | 0.776 ±0.008* | 0.820 ±0.004* | 0.885 ±0.003* | 0.851 ±0.003* |
| ISWC2017 | 0.820 ±0.004* | 0.914 ±0.003* | 0.865 ±0.003* | 0.671 ±0.010* | 0.812 ±0.009* | 0.735 ±0.009* | 0.789 ±0.004* | 0.890 ±0.003 | 0.836 ±0.003* |
| MOST_FOLLOWERS | 0.700 ±0.005* | 0.895 ±0.004* | 0.785 ±0.004* | 0.542 ±0.010* | 0.784 ±0.010* | 0.641 ±0.009* | 0.664 ±0.004* | 0.872 ±0.004* | 0.753 ±0.004* |
| MOST_POPULAR | 0.669 ±0.005* | 0.855 ±0.004* | 0.750 ±0.004* | 0.532 ±0.010* | 0.770 ±0.010* | 0.629 ±0.010* | 0.637 ±0.005* | 0.837 ±0.004* | 0.724 ±0.004* |

Figure 3.7. P/R curves of candidate selection phase: (a) all entities; (b) persons; (c) organizations; (d) precision, recall, and F1 scores for the setting maximizing F1, with confidence intervals and statistical significance (*) of difference wrt. best model

In our experiments and as mentioned in Sections 3.1 and 3.5, the transformation layer turns out to be crucial for successfully training the network. Without it — i.e., with the system variant BASE_KB_SG that simply concatenates the BASE features with the RDF graph (KB) and social graph (SG) embeddings — good models ($P = 0.845 \pm 0.006$, $R = 0.883 \pm 0.006$, $F_1 = 0.863 \pm 0.005$, for all entities) can be trained only for two cross-validation folds out of five, while the models obtained for the other folds are significantly under-performing ($P = 0.744 \pm 0.006$, $R = 0.878 \pm 0.005$, $F_1 = 0.805 \pm 0.005$), bringing down the average performances of BASE_KB_SG over all folds ($P = 0.746 \pm 0.004$, $R = 0.900 \pm 0.003$, $F_1 = 0.815 \pm 0.003$). Even restricting to the two good folds, the performances of BASE_KB_SG there are significantly lower than the ones of BASE_KB_SG_TL, making not worthwhile any attempt at fixing the training issues with the simple concatenation model BASE_KB_SG.

### 3.6.5 Evaluation by Entity and Profile Type

To investigate for which types of entities and profiles the approach performs best, in Table 3.5 we report the performances of candidate acquisition (recall), candidate selection ($P$, $R$, $F_1$), and joint task ($P$, $R$, $F_1$) on different subsets of the gold standard. As candidate selection model, we use the best performing model BASE_KB_SG_TL with the abstention score threshold that maximizes $F_1$ on the whole gold standard. As subsets of the gold standard, we consider:

- person and organization subclasses in DBpedia that have at least 1000 samples in the gold standard (to provide more accurate performance estimates);
- entities whose profile has more vs. less followers on Twitter, using the median number of followers measured on the gold standard as separation value;
- entities whose profile is more vs. less popular in Twitter, with popularity defined as the number of times we observed the profile in the tweet stream and using the median number of occurrences measured on the gold standard as separation value;
- entities with a verified vs. not verified profile.

The table also reports the size of each subset, 95% confidence intervals, and statistical significance of score differences with respect to the whole gold standard population.[12]

[12] We compare the performances of the subset with the ones of its complement using the non-paired approximate randomization test.

Table 3.5. Performances of candidate acquisition, candidate selection (best $F_1$ setting of BASE_KB_SG_TL), and joint task for subsets of the gold standard, with confidence intervals and statistical significance ($^*$) of differences wrt. whole population

| Gold standard subset | # Samples | Candidate acquisition recall | Candidate selection | | | Joint task | | |
|---|---|---|---|---|---|---|---|---|
| | | | P | R | F1 | P | R | F1 |
| dbo:Person | 41003 | 0.652 ±0.005* | 0.865 ±0.004* | 0.917 ±0.003* | 0.890 ±0.003* | 0.845 ±0.004* | 0.608 ±0.005* | 0.707 ±0.004* |
| dbo:Artist | 17764 | 0.617 ±0.007* | 0.861 ±0.006 | 0.919 ±0.005* | 0.889 ±0.005* | 0.836 ±0.007 | 0.578 ±0.007* | 0.683 ±0.007* |
| dbo:Athlete | 10207 | 0.632 ±0.010* | 0.854 ±0.008 | 0.936 ±0.006* | 0.893 ±0.006* | 0.838 ±0.008 | 0.600 ±0.010* | 0.699 ±0.009* |
| dbo:OfficeHolder | 2911 | 0.725 ±0.016* | 0.884 ±0.013* | 0.913 ±0.012* | 0.898 ±0.011* | 0.874 ±0.014* | 0.672 ±0.017* | 0.760 ±0.015* |
| dbo:Politician | 3628 | 0.714 ±0.015* | 0.904 ±0.011* | 0.946 ±0.009* | 0.925 ±0.009* | 0.895 ±0.012* | 0.681 ±0.015* | 0.773 ±0.013* |
| dbo:Writer | 1176 | 0.702 ±0.026* | 0.901 ±0.021* | 0.892 ±0.022 | 0.896 ±0.018* | 0.879 ±0.023* | 0.641 ±0.027* | 0.741 ±0.024* |
| dbo:Organisation | 15175 | 0.461 ±0.008* | 0.821 ±0.009* | 0.800 ±0.010* | 0.810 ±0.008* | 0.775 ±0.009* | 0.391 ±0.008* | 0.520 ±0.009* |
| dbo:Broadcaster | 1054 | 0.493 ±0.030* | 0.778 ±0.040* | 0.633 ±0.041* | 0.698 ±0.036* | 0.702 ±0.040* | 0.337 ±0.029* | 0.455 ±0.032* |
| dbo:Company | 3363 | 0.543 ±0.017* | 0.837 ±0.018* | 0.752 ±0.020* | 0.792 ±0.016* | 0.790 ±0.018* | 0.443 ±0.017* | 0.568 ±0.017* |
| dbo:EducationalInstitution | 1373 | 0.239 ±0.023* | 0.881 ±0.038 | 0.744 ±0.048* | 0.807 ±0.038* | 0.851 ±0.040 | 0.195 ±0.021* | 0.318 ±0.029* |
| dbo:Group | 4705 | 0.529 ±0.014* | 0.819 ±0.014* | 0.903 ±0.012* | 0.859 ±0.011* | 0.769 ±0.015* | 0.489 ±0.014* | 0.598 ±0.014* |
| dbo:SportsTeam | 2313 | 0.513 ±0.020* | 0.802 ±0.022* | 0.806 ±0.023* | 0.804 ±0.019* | 0.775 ±0.023* | 0.440 ±0.021* | 0.561 ±0.022* |
| more followers | 26358 | 0.721 ±0.005* | 0.918 ±0.004* | 0.918 ±0.004* | 0.918 ±0.003* | 0.898 ±0.004* | 0.675 ±0.006* | 0.770 ±0.005* |
| less followers | 26357 | 0.558 ±0.006* | 0.832 ±0.006* | 0.859 ±0.006* | 0.846 ±0.005* | 0.805 ±0.006* | 0.496 ±0.006* | 0.614 ±0.006* |
| more popular | 26367 | 0.718 ±0.005* | 0.919 ±0.004* | 0.913 ±0.004* | 0.916 ±0.003* | 0.899 ±0.004* | 0.668 ±0.006* | 0.767 ±0.005* |
| less popular | 26348 | 0.561 ±0.006* | 0.832 ±0.006* | 0.866 ±0.006* | 0.849 ±0.005* | 0.805 ±0.006* | 0.502 ±0.006* | 0.619 ±0.006* |
| verified | 26194 | 0.755 ±0.005* | 0.932 ±0.004* | 0.939 ±0.003* | 0.936 ±0.003* | 0.915 ±0.004* | 0.719 ±0.005* | 0.805 ±0.004* |
| not verified | 26521 | 0.526 ±0.006* | 0.807 ±0.006* | 0.826 ±0.006* | 0.816 ±0.005* | 0.778 ±0.007* | 0.453 ±0.006* | 0.573 ±0.006* |

Table 3.6. Error breakdown for the joint task using the best performing model (BASE_KB_SG_TL) as evaluated on the gold standard. Abstention counts as an error.

| Error type | Share | Total | Number of per. | Number of org. |
|---|---|---|---|---|
| ALL | 100.00% | 24,477 | 15,566 | 8,911 |
| STREAM | 9.54% | 2,334 | 1,499 | 835 |
| CA_INDEX | 78.57% | 19,233 | 12,316 | 6,917 |
| CA_CUTOFF | 3.60% | 882 | 473 | 409 |
| CS_ABSTAIN | 2.68% | 656 | 349 | 307 |
| CS_DISAMBIG | 5.61% | 1,372 | 929 | 443 |

In terms of person and organization subclasses, Table 3.5 shows that each subset exhibits performances that differ from the average population, with differences always statistically significant for candidate acquisition and $F_1$ of candidate selection and joint task. Among persons, candidate selection performances are similar while lower candidate acquisition recall is observed for artists and athletes (more numerous), which maps to higher recall and thus $F_1$ in the joint task. Among organizations, candidate selection performs better for musical groups and worse for broadcasters, while educational institutions show a significantly low candidate acquisition recall that maps to very low joint recall and $F_1$ scores.

In terms of followers, popularity, and verified status, Table 3.5 shows that entities whose profiles have these characteristics are linked significantly better by our approach. Apart being directly used as candidate selection features, these characteristics generally imply less ambiguity as well as the availability of more abundant and accurate data on the social media side, which ease the linking task.

### 3.6.6 Error Analysis

Here we analyze the types of errors our joint pipeline commits, providing examples and estimating their impact so to inform possible extensions of the approach. We consider as error any answer by the system that differs from the one provided by the gold standard, including system abstention (since the alignment exists). On the whole gold standard, our pipeline with the best performing selection model BASE_KB_SG_TL (best $F_1$ setting) makes a total of $24,477$ errors, originating from all three constituent processing phases. In each of those phases we identify the types of errors, the detailed breakdown of which is provided in Table 3.6.

Firstly, the user index built during the data acquisition phase does not contain the entire population of Twitter. While we can be all but sure that popular entities will end up being in our index, passive users (i.e., the ones that do not tweet and retweet)

will never be observed in the stream of tweets and will never be linked to an entity. For example, at the time of writing, the correct profile for the American music band "The Spares" (dbr:The_Spares), @thespares, has never tweeted or interacted in any way with the rest of the network, therefore will never appear in our user index. Moreover, even when searched via Twitter itself, the correct profile is not returned as one of the options. This type of error, which we identify with STREAM in Table 3.6, is responsible for 9.54% of all errors.

Secondly, two types of errors, CA_INDEX and CA_CUTOFF, originate from the candidate acquisition (CA) phase. CA_INDEX errors amounts to 78.57% of all errors and occur if the profile, while being in the user index, cannot be retrieved by our full-text search candidate acquisition approach. This happens if there is a mismatch between the name contained in DBpedia (which we use for matching) and the name of the Twitter profile, or if the DBpedia name is too ambiguous (i.e., entity known with a popular name, such as John) which will timeout the query and force our candidate acquisition approach to rephrase it. An example of this error occurs with basketball player Walter Tavares (dbr:Walter_Tavares), who is also known as Edy Tavares, which is the name used in his official Twitter profile. This name is not in the English DBpedia and, therefore, his profile would not be returned as a candidate. In this specific case, however, the recall loss can be mitigated by importing names from other DBpedia language chapters or from Wikidata. Concerning CA_CUTOFF errors, they amount to 3.60% of all errors and occur when the correct profile is obtained from the index, but its position in the candidate list is past the cut-off $k = 40$ and it is thus discarded as part of our attempt to reduce potential ambiguity and computational costs.

In third, in case the correct alignment is within the list of candidates, the candidate selection model can commit a CS_ABSTAIN error by wrongly abstaining or commit a CS_DISAMBIG error by selecting the wrong candidate. CS_ABSTAIN errors happen in 2.68% of cases and are caused by the right alignment being rejected as having a score lower than the minimum score threshold (set to 0.169 to maximize $F_1$ on BASE_KB_SG_TL). For example, for the politician and radio host Jason Lewis (dbr:Jason_Lewis_(radio_host)), SocialLink correctly identified his two official Twitter accounts, @Jason2CD and @RepJasonLewis, with the DNN returning very high probabilities, 0.952 and 0.949 respectively. However, since the name has a high degree of ambiguity — there are other American politicians with the same name, not to mention an actor and a comedian, all of which has social media presence — the rescaled scores (see Section 3.5) for those two candidates decrease to 0.132 and 0.131, causing the model to abstain in this case.

CS_DISAMBIG errors represent 5.61% of errors and most frequently result from linking to (i) the profile of a related / topically similar entity, or (ii) an alternative profile of the target entity. An example of the first kind is Santiago Segura (dbr:Santiago_Segura), an actor

and film director, which is aligned to the profile of a namesake who is also an actor. As often the case in such examples, the confidence scores for their respective profiles are very similar (0.952 vs. 0.953), and here, without using image data, even for a human it is hard to select a correct alignment. An example of the second kind is U.C. Sampdoria (`dbr:U.C._Sampdoria`), a football club, which is linked in the gold standard to its official Italian Twitter account `@sampdoria`. SocialLink, on the other hand, links to `@sampdoria_en`, which is the official English account of the same team. In this case, the DNN emits very similar probabilities: 0.712 and 0.728 respectively, which are both above the threshold after rescaling. The third ranked candidate alignment for this entity, `@UCSampdoriaFeed`, has a significantly lower score of 0.473 and is a fan account. Organizations, in particular, have a strong tendency towards having multiple social media profiles targeted at different audiences based on language, market or a group of products. Sometimes it is unclear what the main profile is (if unique), and linking to one of the alternative profiles may not necessarily be a mistake and may justify switching to a 1-to-N problem (and gold standard) where an entity may have multiple corresponding profiles. This, however, does not match the 1-to-1 cardinality of existing links found in both DBpedia and Wikidata and adopted in this thesis.

As can be seen, due to the way the candidate acquisition approach is designed, SocialLink rarely confuses entities with different names but has a number of problematic cases especially when it has to choose between profiles within the same domain or associated to the same target entity. Additionally, when evaluated on the gold standard, SocialLink generally favors precision over recall: the abstention mechanism, designed for the open world case in which we cannot know in advance if the entity has a profile on Twitter, tends to abstain incurring in a false negative error rather than risking to select the wrong candidate and incur in both a false negative and a false positive errors.

## 3.7 On the Choice of Word Embeddings

Pre-trained word representations, or embeddings, have become a staple technique for modeling textual data in a convenient low-dimensional form. Such word representations are typically allocated in the resulting vector space according to the distributional semantics hypothesis, i.e., the words that appear in similar contexts tend to have similar meanings and will be placed close to each other. Pre-trained word representations allow the usage of large unsupervised corpora to model the target languages efficiently. Over the last five years, since the introduction of the word2vec algorithm, many new approaches were proposed to improve different aspects of such representations yielding better performance than the original word2vec in many tasks. Before word2vec, methods, such as LSA, HAL and autoencoders were also widely used. However, to the best of our knowledge at the time

Table 3.7. Precision, recall, F1 scores with the setting maximizing F1 for approaches using different types of embeddings with confidence intervals and statistical significance (*) wrt. `ALL` model

| Embeddings used | P | R | F1 |
|---|---|---|---|
| `ALL` | 0.854 ±0.004 | 0.880 ±0.003 | 0.867 ±0.003 |
| `LSA` | 0.842 ±0.004* | 0.884 ±0.003* | 0.862 ±0.003* |
| `GloVe` | 0.849 ±0.004* | 0.870 ±0.004* | 0.859 ±0.003* |
| `fastText` | 0.842 ±0.004* | 0.877 ±0.004* | 0.859 ±0.003* |

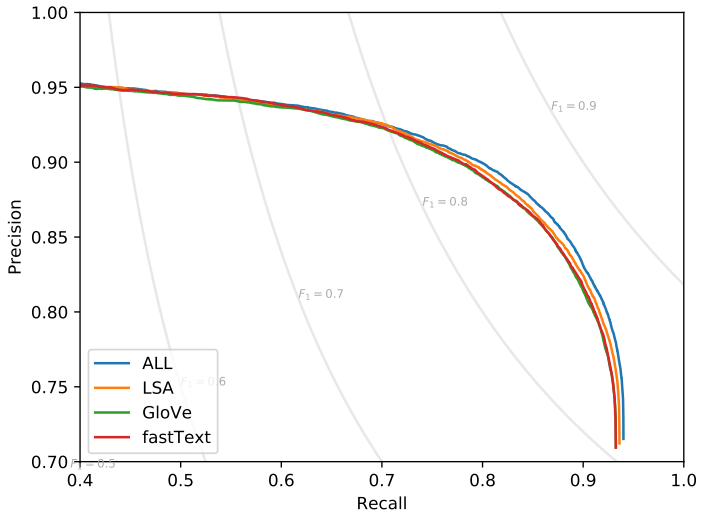



Figure 3.8. P/R curves of four embedding combinations using the BASE_KB_SG_TL model

of writing, there is still no consensus in the community about whether any of the proposed approaches is clearly superior to others and should be used by default to represent text. This conclusion is consistent with the famous "no free lunch" concept, meaning that for each task the choice of particular word embeddings should be justified and supported with experiments.

Here we measure the impact of different pre-trained word representations on our task. As mentioned in Section 3.3.1, we chose the Latent Semantic Analysis (LSA) approach to represent text. The choice was mainly driven by our confidence in our LSA-based model that was used in DBpedia-related tasks before. In this section, we conduct an additional set of experiments in order to measure the impact of the choice of different embeddings on our task. This will not only serve as an extra data point for the community but will also potentially allow us to improve the performance of our approach in the future.

### 3.7.1 Experimental Setting

We compare the previously chosen LSA model to two recent embedding types: GloVe (Pennington et al., 2014) and `fastText` (Bojanowski et al., 2017). To this end, we modify the computation of the "Description" scalars in our original BASE feature set (see Table 3.3). Each scalar is computed as follows. Each user text and entity text is converted into the sparse vector $\mathbf{x}_{\text{sparse}} \in \mathbb{R}^v$, where $v$ is the size of the vocabulary for the given language model. Each vector contains tf-idf scores for each token $t$ present in a text:

$$x_t = \text{tf}(t) \cdot \text{idf}(t, D) = \log(1 + \text{freq}_t) \cdot \log\left(1 + \frac{|D|}{1 + |\{d \in D : t \in d\}|}\right)$$

where D is a corpus on which a chosen language model was trained. As can be seen, the computation of such vector requires IDF statistics from the corpus. Here we use precomputed vectors provided by the authors of the respective approaches which do not include such data along with the embeddings. Therefore, we use the same IDF scores acquired from Wikipedia for all models. Each approach is represented by the embedding matrix $M \in \mathbb{R}^{v \times d}$, where $d$ is the embedding size. Then the dense text vector is acquired: $\mathbf{x}_{\text{dense}} = \mathbf{x}^T_{\text{sparse}} \cdot M$. Finally, the Description scalars are computed as a cosine similarity between the user text $\mathbf{u}_{\text{dense}}$ and the entity text $\mathbf{e}_{\text{dense}}$. In short, in order to test other representation approaches we substitute the embedding matrix $M_{\text{lsa}}$ we employed originally with $M_{\texttt{fastText}}$ and $M_{\text{GloVe}}$.

We test four different word embedding models and measure their impact on the performance of the BASE_KB_SG_TL approach described in the chapter:

1. `LSA` ($v = 972,001; d = 100$). The same LSA-based approach we used throughout the chapter. The model is derived from Wikipedia and is described in Aprosio et al. (2013).

2. `GloVe` ($v = 1,917,494; d = 300$). The model is trained on the 42B token Common Crawl corpus and provided by the authors on their website.[13] This approach is also described in Chapter 2 and used to populate the RDF embeddings we used.

3. `fastText` ($v = 2,519,370; d = 300$). The word2vec-based model exploiting subword information. The model was provided[14] by the authors and is trained on Wikipedia.

4. `ALL`. Description scalars produced by each model used together. Effectively an ensemble of embeddings.

[13] http://nlp.stanford.edu/data/glove.42B.300d.zip
[14] https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

Additionally, we slightly modify our *data acquisition* phase. During this phase, we would typically gather the textual content for each candidate from the stream of tweets. As a consequence, we would have a lot of textual content for more popular users and much less (as little as a description field in the profile) for others. This makes it so the "Description" features get a strong implicit notion of the user popularity instead of capturing only the textual similarity. To alleviate this bias, we freshly captured the most recent 200 tweets for each candidate for each entity in our gold standard using Twitter REST API.

All approaches are evaluated using stratified 5-fold cross validation, additionally computing 95% confidence intervals and performing statistical significance test using the approximate randomization test (see Section 3.6.1).

### 3.7.2 Experimental Results

Evaluation results for the four models are provided in Table 3.7, while Figure 3.8 provides the precision / recall curves. As can be seen, the difference between the four models is minimal. However, the `ALL` model does provide statistically significant improvement over the `LSA` model we employed originally, and similarly over `fastText` and `GloVe`.

The absence of significant differences between separate models shows that our pipeline is not particularly sensitive towards the choice of the particular word embeddings. In future, the model can be modified to include textual representations in a similar way we have incorporated the graphical ones, to give more opportunity for the neural network to utilize textual data efficiently.

## 3.8 Related Work

Recently, some researchers have tried to link social media accounts to Wikipedia categories as a way of inferring user attributes, for example, interests. This usually implies linking such accounts to Wikipedia articles that are in turn closely related to entities in the Wikipedia-based KBs, such as DBpedia, YAGO and Wikidata. Faralli et al. (2015a) presents an approach that links Twitter profiles to Wikipedia pages using BabelNet and Babelfy and then to top 22 Wikipedia categories to determine a target user's or community's category. Besel et al. (2016) introduced a similar interest inference pipeline using MediaWiki Web API and used a spreading activation technique on the Wikipedia Bitaxonomy to acquire interests. Piao and Breslin (2017) iterated on this idea improving various steps of the pipeline. In this thesis (see Chapter 6), we have also used an earlier version ($v2.0$, as described in Chapter 4) of SocialLink to perform the linking and then propose a list of users to follow to conceal target user's interests as a way to defend against the inference approaches listed above. In these pipelines, the performance of the

linking approach is not as important. The recall is mostly irrelevant in this case, and incorrect disambiguation of a Wikipedia page does not matter if it acquires correct or similar categories in the end. In our case, we aim to disambiguate as best as we can for any given recall level, which makes SocialLink suitable for a wider array of tasks.

To the best of our knowledge, no one has tried to link social media profiles directly to KB entries. However, this task is closely related to the profile matching (or profile aligning) problem, whose goal is to align profiles of the same person in different social media. A lot of research has been done for this task, exploiting various attributes in profiles (Peled et al., 2016; Goga et al., 2015; Lu et al., 2014), user-generated content (Peled et al., 2016; Liu et al., 2014; Goga et al., 2013), and social graphs (Lu et al., 2014).

Some researchers have pointed out that most of the attributes that could theoretically be exposed in social media are unreliable for profile matching (Goga et al., 2015). Attributes might not exist, might contain information of varying granularity, or they might even be false. Attempts were therefore made to use as little information as possible to align profiles, choosing only the most reliable attributes. In Zafarani and Liu (2013); Zafarani and Liu (2009) only the username (which is the unique identifier that exists in virtually any social media) was exploited to align profiles. The authors showed that people tend to be very consistent when choosing their usernames, which enables identification even if the rest of the profile is filled with incorrect information. Even though they proved that username is a powerful feature, KBs typically do not contain examples of usernames, which makes this feature unavailable for our task.

Goga et al. (2015) explored the reliability of attributes in various social media. According to their study, only few attributes like username and real name are available in social media reliably. For example, they reported that location is present in 54% of Twitter profiles and is not consistent across multiple social media. They exposed various methodological and technical challenges in this area related to the construction of ground truth datasets, attribute discriminability and impersonability. Goga's PhD thesis (Goga, 2014) provides a more in-depth look into those issues.

Liu et al. (2014) reported the largest experiment on profile matching to date using a dataset of 10 millions profiles across 7 social media. Their approach leverages a wide variety of hand-crafted features based on textual and image user-generated content, and demonstrates its importance for profile matching. Note that user-generated content is usually missing in KBs, so it cannot be used in our task as it is used in the profile alignment task.

Goga et al. (2013) showed that profiles can be matched robustly even if explicit attributes are hidden or intentionally falsified. Their approach uses only implicit features present in social media but generally unavailable in KBs, such as writing style, messaging behavior, and location metadata.

Social graph has proven to be hard to acquire in social media. It could be unavailable for crawling or there could be very strict restrictions on API (most notably, in Twitter). Lu et al. (2014) were able to gather a small social graph dataset and proved that it can be effectively matched to improve the results of profile alignment. Entities in KBs often contain links to other entities which can be interpreted as a kind of social graph.

Finally, Peled et al. (2016) gave an overview of the profile alignment task and presented their own approach that uses all available information in the profile to perform matching. They presented three main use cases for their system, one of which — searching for a user by similar name — is close to the candidate acquisition part of our system.

To summarize, even though some researchers expressed concerns (Goga et al., 2015; Goga, 2014) regarding the use of some attributes, every piece of profile data contributes towards identifying the user.

In this chapter, we leverage and extend graph embeddings (or network embeddings), a particular type of embeddings that is an intensely researched topic in recent literature (Goyal and Ferrara, 2017). We refer reader to Chapter 2, where we have provided an extensive overview of such approaches including the ones used in this chapter.

## 3.9 Conclusions and Future Work

In this chapter, we presented our supervised approach for automatically linking KB entities to corresponding social media profiles, which we apply and evaluate on DBpedia and Twitter. We have contributed a novel three-phase linking approach to perform such linking. Firstly, we introduce user and entity indices as a way to preprocess large quantities of data. Secondly, we devise a variety of strategies to acquire a subset of candidate alignments for each entity from the KB. Finally, we design an efficient feature set and the deep learning-based approach that is able to link an entity to its corresponding profile. Our BASE feature set provides similarity measures based on various textual, categorical and numerical data extracted from both the target entity and candidate social profile.

In its most complete form, our approach allows leveraging graph-based features both on the social media and KB sides, encoding them in the form of embeddings, i.e., low-dimensional representations of objects trained from massive amounts of unlabeled data, that we gather from both the DBpedia RDF graph and the Twitter social graph.

For the social graph embeddings, we devised a custom procedure to estimate the social graph from the publicly observable stream of tweets, followed by the application of the co-occurrence matrix factorization method called Swivel. We showed that it is possible to acquire an approximation of the social graph indirectly and calculate embeddings at scale.

For the RDF graph embeddings, we used the precomputed embeddings provided by Cochez et al. (2017b). Testing those embeddings on our task, we confirmed the findings of

their authors about the significance and impact on performances of the weighting schema adopted in their production when applying such representations to a particular problem.

Additionally, we found that the combination of hand-crafted features and pre-trained embeddings in a prediction task like ours requires significant redesign of the learning approach. By changing the topology of our neural network, we were able to achieve a stable performance improvement from the adoption of the new features.

Finally, we have evaluated in detail the entire pipeline with the specific focus on the effect that graph-based features and the modified neural architecture produce on our approach. We observed significant improvements when applying graph-based features derived from DBpedia and Twitter separately, which were still dominated by the complete model. In addition, we provide an extensive error analysis of all the phases of our linking process including separate set of experiments justifying the choice of a particular text representation approach.

# Chapter 4

# SocialLink Resource

In this chapter, we present a second core contribution of this thesis — the SocialLink dataset, a publicly available Linked Open Data dataset that contains links between the social media accounts on Twitter and the corresponding entities in multiple language chapters of DBpedia. By following the approach presented in Chapter 3 we are able to produce a significant number of alignments to enable the abovementioned knowledge transfer between the two worlds. As a result, on the one hand, we support Semantic Web practitioners in better harvesting the vast amounts of valuable, up-to-date information available in Twitter; on the other hand, the resource presented here permits Social Media researchers to leverage DBpedia data with little effort when processing the noisy, semi-structured data of Twitter.

This resource is part of the SocialLink project and is periodically updated with releases. The code along with the gold standard dataset used to produce it are made available as part of our open source project. This chapter contains details on the design choices, various dataset statistics and provides discussion about some of the general use cases that showcase the above-mentioned knowledge transfer.

## 4.1 Introduction

The number of existing links to social media for the living people and organizations in the Linked Open Data cloud is very low: the DBpedia 2016-04 that we currently employ in our experiments contains just 56,133 of them. In order to enable the knowledge transfer between the LOD and the social media, the coverage has to be significantly improved. In Chapter 3, we have introduced a scalable approach that can potentially help to fill this gap by providing a significant amount of high-quality links automatically, learning from the existing ones.

In this chapter, we present the SocialLink dataset,[1] a publicly available Linked Open Data dataset based on our state-of-the-art linking approach that matches social media accounts on Twitter to their corresponding entities in DBpedia. Over the last two years we have released three major versions of this dataset. The latest version, $v3.0$, consists of almost 322K high quality (more than 90% precision) alignments, obtained by applying the above linking approach to 2M living people and 500K currently existing organizations in DBpedia. Entities from 128 DBpedia language chapters are considered: while the textual features we extract are mainly designed to work with Western languages, the linking approach employs a wide variety of different feature types allowing us to target entities in other languages as well. Additionally, the dataset contains raw scores for each candidate alignment allowing end users to tune the precision / recall balance as they see fit, potentially obtaining up to 1M of alignments. The dataset is available in many different formats including RDF/OWL, which we then distribute in accordance with LOD best practices (Wilkinson et al., 2016), reusing existing vocabularies and providing a live SPARQL endpoint.

By covering such amount of entities, SocialLink dataset indeed creates a bridge between the highly structured LOD cloud and the vibrant and up-to-date social media world. SocialLink dataset serves two main purposes. On the one hand, it aims at facilitating social media processing by leveraging DBpedia data, e.g., as a source of ground truth properties for training supervised systems for user profiling, or as contextual data in natural language understanding tasks (e.g., Named Entity Linking) operating on social media contents (Corcoglioniti et al., 2016; Minard et al., 2016). On the other hand, SocialLink gives Semantic Web practitioners the ability to populate KBs with up-to-date data from social media accounts of DBpedia entities, such as structured attributes, images, connections, user locations, and descriptions. To the best of our knowledge, SocialLink dataset is unique in the alignment task it addresses providing more than tenfold increase in the number of links to Twitter in the LOD.

SocialLink project focuses on linking living people and organizations. There are two main reasons to that. Firstly, those entities of those types constitute the vast majority of entities that can be reasonably aligned to a social media profile. Indeed, social media were designed to accommodate just living people initially. With such a lucrative commercial opportunity that social media represent, various organizations started to cultivate presence there for themselves too. Both public people and organizations use social media to engage with their audience and potential customers, share relevant information and promote their products and services. Secondly, such entity types constitute the overwhelming majority of existing links between DBpedia and Twitter. While SocialLink approach does support other

[1] http://w3id.org/sociallink — Creative Commons Attribution license (CC BY 4.0).

entity types (our preprocessing pipeline and features are entity type-agnostic), as shown in Chapter 3, there are vast differences in linking quality based on entity type: linking of organizations compared to persons exhibit up to 18% lower $F_1$ using the same approach. The absence of the significant number of training samples will inevitably strengthen this issue. Because of that we decided to limit the scope of the resource to only two entity types.

The first version of SocialLink dataset, $v0.1$-alpha, was released in mid 2016 using the supervised alignment approach described in Nechaev et al. (2017b). Since then, we have significantly expanded its scope and alleviated some of the restrictions of the original system. To name a few, the approach is no longer restricted by the limits of Twitter REST API and is now able to use entity data from 128 DBpedia chapters, allowing us to align DBpedia entities present only in localized DBpedia chapters, and to provide more context to our matching algorithm, helping with disambiguation and increasing the amount of processed entities by a factor of three. The SocialLink pipeline generating the dataset is available open source[2] along with the revised gold standard dataset used to train and evaluate the system. We are able to repopulate the dataset in an automatic way covering the latest data and algorithm improvements to insure that alignments are up-to-date. Relevant statistics and the latest dataset version can be found on our website and Zenodo.[3] At the time of writing, we have released three major versions of the resource, each corresponding to a major milestone in SocialLink development (Nechaev et al., 2017b,d, 2018b). The SPARQL endpoint available on our website always contains the latest public release of our resource.

In the remainder of the chapter, Sections 4.2 and 4.3 present respectively the SocialLink pipeline used to produce the resource and the latest version of the SocialLink dataset. Section 4.4 discusses some of the scenarios where the dataset has been or can be used, while Section 4.5 concludes.

## 4.2 SocialLink Pipeline

The SocialLink dataset population procedure follows the same three-phase pipeline described in Chapter 3. Briefly, processing starts with the *data acquisition* phase, where the required Twitter and DBpedia data, including preexisting gold standard alignments from DBpedia, are gathered, prepared and indexed locally for further processing. Next, in the *candidate acquisition* phase, for each DBpedia entity, a list of candidate matching Twitter profiles is obtained by querying the indexes. Finally, the *candidate selection* phase uses the gold standard alignments to train a Deep Neural Network (DNN) that scores and selects the

[2] http://github.com/Remper/sociallink
[3] https://doi.org/10.5281/zenodo.1451797

best matching candidate. The system may abstain if there is no suitable candidate. After an entity passes through this pipeline, it is ready to be added to the resource.

In this section, we provide additional details on the linking approach presented in Chapter 3 relevant to the population of the resource. Namely, we discuss coverage issues with some of the feature families, provide details on employing the SocialLink approach for large multi-language KBs and highlight pipeline differences between different versions of the SocialLink dataset. In total, there has been five releases of the resource available on our website[4] including the three major ones covered in the respective publications. The first two versions, $v0.1$-alpha and $v0.5$-beta, were produced during initial experiments and implementation of the base pipeline. The first major version, $v1.0$, uses the approach described in Nechaev et al. (2017b) and is the first release available on Zenodo. The $v2.0$ covers improvements made in Nechaev et al. (2017d), while the most recent release, $v3.0$, uses the updated approach from Nechaev et al. (2018b). In this section, we will refer to $v1.0$ and $v2.0$ as Legacy and $v3.0$ as Current.

### 4.2.1 Feature Coverage

The Legacy versions of SocialLink employ a simplified BASE feature set, while Current version benefit from the latest BASE_KB_SG_TL system (see Section 3.5). The introduction of graph-based features to Current has significantly improved the performance of our linking approach. However, when applied to entities outside of the gold standard, the coverage aspect of the new features has to be considered. While our Knowledge Base embeddings (Cochez et al., 2017b) cover almost 9M entities of DBpedia, they consider only the English chapter, while the SocialLink dataset covers 128 different language chapters. This yielded $68,9\%$ coverage during population of Current. For the experiments described in this thesis, we consider such coverage sufficient. However, in future a joint embedding can be trained exploiting owl:sameAs links across language chapters to avoid significant drops in coverage. The same can be observed in social media: we approximate the social graph from the incomplete stream of tweets, inevitably losing perfect coverage. The latest version of our approximated social graph contains data for 168M users which covers $65,3\%$ of candidates in Current. Social graph embedding is acquired from the precomputed embeddings of friends, so the process fails if and only if the target user hasn't been observed interacting with any of the known users as sampled from Twitter Streaming API. When we calculate graph-based features, in case where the graph-based features are not available, we default to an average embedding vector for the respective subspaces: the average of all entity embeddings from the KB side and the average of the most popular users from the social media side.

[4] http://w3id.org/sociallink#download

Our BASE (see Table 3.3) feature family can theoretically include features with imperfect coverage as well. For instance, textual features, such as similarities based on names, descriptions and tweets, and profile features default to zeros at training and evaluation time in case the profile object for the candidate was not found in our index. However, for the resource population, we exclude all candidates that did not have at least a profile object in our database always providing at least a minimal textual context and important structured attributes, such as names. The *homepage links* feature family containing 327K candidate-entity pairs has limited coverage as well: it was crawled in mid 2016 based on entities from 2015-10 edition of DBpedia and will become more stale as the time goes by. As this feature family is hard to update and it introduces limited performance improvements, we plan to replace it with a more scalable set of features based on machine-readable properties in next versions of SocialLink. For example, feature families described in Chapter 5 can be employed instead.

Besides the explicit coverage issues, text-based features can discard the data if the input does not match the vocabulary. Firstly, the input tokens are produced based on a simple multi-language stemming and tokenization procedure. It performs well on most Western languages but is weak when applied to Asian and Arabic languages. Secondly, the vocabulary of the produced tokens itself is limited. In Legacy versions, the LSA-based model was employed having $972,001$ most frequent tokens as seen in six-language Wikipedia-based corpus (Aprosio et al., 2013). In Current, we employ three different language models described in Section 3.7, which significantly expands the available vocabulary. However, its performance is still heavily biased towards English and similar languages.

### 4.2.2 Scoring and Selection Procedures

During evaluation of the SocialLink approach, we employ five-fold cross-validation to provide accurate assessment of the algorithm performance. In order to produce the resource, we acquire pairwise (entity-candidate) predictions from each fold. Then, to ensure stable predictions, we implement a basic ensemble of the models from each fold by averaging the pairwise scores across all folds. After this averaging procedure, for each entity the algorithm has to either select the best suitable candidate (candidate with the largest score) or abstain from selection. To do that, Legacy versions employed two predefined thresholds: *minimum score* required to consider an alignment correct and *minimum improvement* over the second best pick. The latter ensures that the algorithm can abstain if two or more candidates are indistinguishable, even if they pass the minimum score requirement. The thresholds can be selected based on the precision/recall curve produced from the evaluation on a gold standard as shown in Figure 3.4.

For example, for the $v2.0$ release, the thresholds were optimized for the desired precision of 90% and set to 0.4 minimum score and 0.4 minimum improvement, leading to 41% recall of generated alignments (Nechaev et al., 2017d). For the older $v1.0$ release, the F1-optimized setting was used yielding 85% precision and 52% recall (Nechaev et al., 2017b). Since different downstream tasks can impose different precision requirements from alignments, we always provide the raw scores for each entity to enable the user to select appropriate thresholds based on those requirements.

The Current ($v3.0$) release employs an alternative strategy described in Section 3.5 alleviating the need for the *minimum improvement* threshold by rescaling the scores to a proper probability-like distribution considering each candidate plus abstention as possible outcomes. The rescaling approach heavily punishes the scores for entities that are highly ambiguous and has almost no effect in cases where there is a single good choice. The minimum score threshold is still selected based on the evaluation on a gold standard but has one less parameter to tune. As in $v2.0$, we have selected 90% precision target corresponding to 0.28 threshold yielding 55% recall (with 322 307 total alignments) this time due to algorithm and pipeline improvements we have introduced. Have we selected the 0.4 threshold of $v2.0$ for Current release, we would have aligned 248 349 entities corresponding to expected 93% precision and 51% recall. While the new scoring procedure is much more conservative than the previous one leading to more abstentions on ambiguous cases, it simplifies the choice of threshold for the end user and provides more reliable alignments overall.

### 4.2.3 Populating the Resource

The population of the resource follows the general workflow of the SocialLink pipeline. First, the updated data from the social media has to be consolidated. This data (3.3TB at the time of writing) is updated several times during the year and is indexed using a number of Apache Flink[5] pipelines. The resulting index is stored in a PostgreSQL database (747GB). The database schema is available in our repository[6]. Due to Twitter terms of use and privacy concerns we release neither the raw Twitter Streaming API data we used nor the preprocessed index. The user index population typically takes six days to fully complete on our hardware (single Xeon E5-2630v4 with 192GB of RAM). From the KB side, we exploit the same DBpedia index detailed in Section 3.3.1.

Then the resource is produced by first computing the list of candidates for each entity and then scoring each pair independently. The Current version has 14 011 602 of such pairs. Then, based on the selected threshold, the resource is produced in different formats

[5] https://flink.apache.org/

[6] https://github.com/Remper/sociallink/blob/master/alignments/src/main/resources/schema.sql

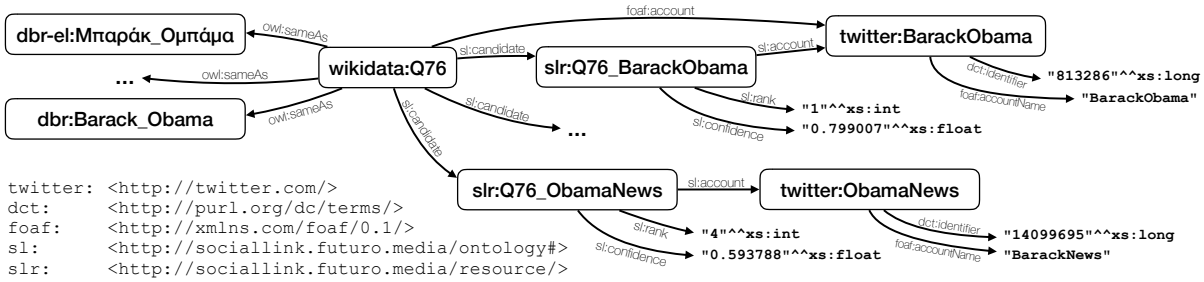



Figure 4.1. Representation of alignments in RDF.

detailed in Section 4.3 and on our website. The code for generating the dataset is written in Java and Python as part of our open source project and the documentation for running the pipelines is available online.[7] Additionally, in Chapter 7, we discuss Social Media Toolkit (SMT). Complementary to its entity linking capabilities, SMT acts as a web-based test bench allowing us to debug and validate the resulting SocialLink dataset and to deploy it in production via the API.

## 4.3 SocialLink Dataset

As mentioned before, the result of running the SocialLink pipeline is the SocialLink dataset, that we are able to generate periodically to account for algorithm updates and new data in DBpedia and Twitter. The dataset is distributed in different formats, with RDF being the main one, including the high quality alignments as well as all the intermediate candidate data. We describe here the modeling choices behind the RDF format of the SocialLink dataset, summarizing the statistics of its three main releases ($v1.0$, $v2.0$, $v3.0$) and discussing how the dataset is made available online and kept up-to-date.

### 4.3.1 RDF Format

We encode our alignments in RDF using terms from FOAF,[8] Dublin Core Terms,[9] and our custom SocialLink vocabulary (prefix `sl`),[10] as exemplified in Figure 4.1.

DBpedia entities are referenced using canonical URIs possibly taken from Wikidata, like wikidata:Q76 for entity Barack Obama. Each canonical URI has owl:sameAs links to itself and to corresponding URIs in other DBpedia chapters (based on gathered DBpedia data), allowing querying the dataset using localized entity URIs.

[7] https://github.com/Remper/sociallink/wiki/align
[8] http://xmlns.com/foaf/0.1/ (prefix: `foaf`).
[9] http://purl.org/dc/terms/ (prefix: `dct`).
[10] http://sociallink.futuro.media/ontology# (prefix: `sl`).

Table 4.1. Number of entities considered in different versions of SocialLink.

| Entity type | Entities in DBpedia | | Entities considered (per version) | |
|---|---|---|---|---|
| | 2015-10$_{\text{en only}}$ | 2016-04$_{\text{multi-lang}}$ | $v1.0$ | $v2.0$ / $v3.0$ |
| dbo:Person | 1 365 651 | 2 975 645 | 702 530 (51.4%) | 2 035 590 (68.4%) |
| dbo:Athlete | 372 424 | 493 867 | 252 268 (67.7%) | 412 629 (83.6%) |
| dbo:Artist | 146 759 | 269 745 | 84 147 (57.3%) | 188 095 (69.7%) |
| dbo:Politician | 67 074 | 123 460 | 29 750 (44.4%) | 65 135 (52.8%) |
| dbo:Writer | 41 978 | 69 753 | 19 055 (45.4%) | 37 744 (54.1%) |
| dbo:Scientist | 35 851 | 64 005 | 13 634 (38.0%) | 28 854 (45.0%) |
| dbo:SportsManager | 17 857 | 18 281 | 13 477 (75.7%) | 14 860 (81.3%) |
| dbo:Coach | 8 452 | 8 772 | 5 142 (60.8%) | 5 643 (64.3%) |
| dbo:Model | 2 824 | 7 601 | 2 470 (87.5%) | 7 470 (98.3%) |
| dbo:Journalist | 2 331 | 4 019 | 1 647 (70.7%) | 3 285 (81.7%) |
| dbo:Presenter | 821 | 4 898 | 643 (78.3%) | 4 674 (95.4%) |
| dbo:Organisation | 346 083 | 575 644 | 171 187 (49.5%) | 531 177 (92.3%) |
| dbo:Company | 75 398 | 131 056 | 45 384 (60.2%) | 119 365 (91.1%) |
| dbo:EducationalInst. | 62 407 | 116 139 | 28 936 (46.4%) | 108 353 (93.3%) |
| dbo:Group | 42 056 | 66 868 | 31 938 (75.9%) | 61 620 (92.2%) |
| dbo:SportsTeam | 41 227 | 62 221 | 23 107 (56.1%) | 59 257 (95.2%) |
| dbo:Broadcaster | 29 595 | 35 394 | 13 151 (44.4%) | 35 028 (99.0%) |
| dbo:MilitaryUnit | 19 673 | 36 151 | 13 243 (67.3%) | 34 574 (95.6%) |
| dbo:PoliticalParty | 12 480 | 16 611 | 2 367 (19.0%) | 13 767 (82.9%) |
| dbo:GovernmentAgency | 8 461 | 9 634 | 1 654 (19.6%) | 8 647 (89.8%) |
| dbo:Non-ProfitOrg. | 6 035 | 8 109 | 2 354 (39.0%) | 7 772 (95.8%) |
| dbo:TradeUnion | 1 773 | 2 032 | 153 ( 8.6%) | 2 027 (99.8%) |
| All entities | 1 711 734 | 3 551 289 | 873 717 (51.0%) | 2 566 767 (72.3%) |

Twitter accounts, like twitter:BarackObama, are modeled as foaf:OnlineAccount individuals, using properties foaf:accountName and dct:identifier to respectively encode the account screen name and numeric identifier (useful in applications).

The alignment between a DBpedia entity and the corresponding Twitter account is expressed using property foaf:account. In addition, individuals of type sl:Candidate (e.g., slr:Q76_BarackObama in Figure 4.1) reify the many-to-many relation between DBpedia entities and candidate Twitter accounts, linked via properties sl:candidate and sl:account. This reified relation is enriched with properties sl:confidence and sl:rank encoding the candidate confidence score (i.e., estimated correctness probability) and its rank among the candidates for the entity, to simplify querying for the top candidate.

Based on this modeling, the following SPARQL query retrieves the Twitter account (if any) aligned to an entity identified by any of its localized DBpedia URIs <E>:

```
SELECT ?account {?e owl:sameAs <E>; foaf:onlineAccount ?account}
```

### 4.3.2 Dataset Statistics

Table 4.1 reports the number of entities we have considered over time. In the first main release of the SocialLink dataset, $v1.0$, only English DBpedia (release 2015-10) was considered, providing 1 771 734 potential entities of types dbo:Person and Organization to align. Given our definition of live person and organization at the time, we have filtered out 49% of those entities ending up with just 873 717 potential targets for alignment. For subsequent releases, including the latest one, we have significantly expanded the scope of the SocialLink dataset taking 128 DBpedia language chapters of the 2016-04 release as the source. The usage of multi-lingual input has increased the number of target entities threefold. It is worth mentioning that we have also improved the filtering conditions employed to discard non-alive entities, which is the main reason behind the percentage-wise increase in considered entities versus the total (from 51% to 72,3%). Another reason is the coverage issue for smaller chapters of DBpedia where for many entities the properties that would indicate its "alive" status are not filled. Additionally, Table 4.1 indicates the number of entities considered belonging to some of the most populous types: for persons we mainly consider athletes, for organizations it is companies.

More importantly, each release of the SocialLink dataset provided different amounts of data. Table 4.2 showcases this difference by detailing (i) the number of entities with candidates, (ii) average number of candidates per entity and (iii) the number of high quality alignments produced for each version of SocialLink dataset. In the first release, $v1.0$, just over 620K entities had at least one candidate, out of which 304K high quality alignments were produced. The redesigned pipeline of $v2.0$ relying on our custom-built index, expanded list of target entities and the refined approach was able to provide candidates for 906K entities. While it is larger than $v1.0$, percentage-wise from the number of considered entities, it is significantly lower (from 71% of all entities to just 40%). This is mainly due to candidate acquisition being more precise in populating the candidates: Twitter API search used in $v1.0$ would try to respond with at least some candidates even if they are unlikely to be a match based on a name, while our user index would require at least a partial name match. Finally, $v2.0$ contains just under 272K high quality alignments, which is an 11% decrease from $v1.0$. This is mainly due to a more strict precision target (90% compared to 85% of $v1.0$ as discussed in Section 4.2) and a more conservative approach overall. Our latest release, version 3.0, improves the overall quality of the pipeline by introducing algorithm improvements at each phase detailed in Chapter 3. This enabled both the increased number of entities with candidates and the increase of 18,5% in high quality alignments (322 124) compared to $v2.0$ without changing the precision requirements.

### 4.3.3 Availability and Sustainability

The SocialLink dataset is indexed on DataHub[11] and is available for download on SocialLink website, together with VOID (Alexander et al., 2009) statistics, old dataset releases, the gold standard (encoded using the same RDF representation), and non-RDF versions of alignments (JSON, TSV, no intermediate candidate data). Canonical citations (DOIs) for the dataset are available via Springer Nature (Nechaev et al., 2017c) ($v2.0$) and Zenodo (Nechaev et al., 2018a) (all releases) digital repositories. Alignment data is also available and queryable by end users and applications via a publicly accessible SPARQL endpoint[12] using Virtuoso. The SocialLink vocabulary is published according to LOD best practices, and both vocabulary and data URIs are dereferenceable with support of content negotiation.

Extensive documentation is available via the website, covering: (i) dataset scope, format, statistics, and access mechanisms; (ii) instructions for deploying and running the SocialLink pipeline to recreate the resource; (iii) example applications using the dataset; and, (iv) links to external resources like the GitHub repository and issue tracker.

The main requirement for generating the SocialLink dataset is the collection of (at least) some months of raw data from the Twitter Streaming API, e.g., via our data acquisition components. We run a SocialLink pipeline on our premises to continuously collect this data and sustain the periodic update of the dataset. No code modifications are foreseen unless breaking changes occurs in formats and APIs of Twitter and DBpedia.

[11] http://datahub.io/dataset/sociallink
[12] http://sociallink.futuro.media/sparql

Table 4.2. Alignment statistics in different versions of SocialLink. Percentages are calculated from the number of entities considered for each version as reported in Table 4.1.

| Entity type | Entities with candidates | | | Cand. / entity | | | Alignments produced | | |
|---|---|---|---|---|---|---|---|---|---|
| | $v1.0$ | $v2.0$ | $v3.0$ | $v1.0$ | $v2.0$ | $v3.0$ | $v1.0$ | $v2.0$ | $v3.0$ |
| dbo:Person | 524 251 (74,6%) | 737 017 (36.2%) | 836 490 (41.1%) | 7.0 | 12.6 | 13.5 | 246 732 (35.1%) | 234 450 (11.5%) | 260 628 (12.8%) |
| dbo:Athlete | 187 748 (74.4%) | 214 070 (51.9%) | 231 036 (56.0%) | 7.0 | 15.1 | 15.4 | 80 727 (32.0%) | 71 935 (17.4%) | 67 266 (16.3%) |
| dbo:Artist | 72 242 (85.9%) | 104 614 (55.6%) | 116 260 (61.8%) | 7.4 | 12.3 | 14.8 | 43 022 (51.1%) | 41 740 (22.2%) | 48 393 (25.7%) |
| dbo:Politician | 21 223 (71.3%) | 28 554 (43.8%) | 31 569 (48.5%) | 6.4 | 11.7 | 10.8 | 11 049 (37.1%) | 12 400 (19.0%) | 14 934 (22.9%) |
| dbo:Writer | 14 758 (77.5%) | 16 630 (44.1%) | 18 904 (50.1%) | 6.6 | 9.6 | 11.9 | 7 912 (41.5%) | 5 195 (13.8%) | 8 020 (21.2%) |
| dbo:Scientist | 8 505 (62.4%) | 6 659 (23.1%) | 7 982 (27.7%) | 6.1 | 9.4 | 11.2 | 3 345 (24.5%) | 1 489 ( 5.2%) | 2 374 ( 8.2%) |
| dbo:SportsManager | 9 509 (70.6%) | 7 409 (49.9%) | 8 396 (56.5%) | 6.9 | 13.9 | 15.8 | 3 492 (25.9%) | 1 708 (11.5%) | 1 953 (13.1%) |
| dbo:Coach | 4 528 (88.1%) | 3 947 (70.0%) | 4 315 (76.5%) | 7.7 | 13.2 | 16.7 | 2 407 (46.8%) | 1 334 (23.6%) | 1 695 (30.0%) |
| dbo:Model | 2 239 (90.7%) | 4 915 (65.8%) | 5 540 (74.2%) | 7.3 | 8.4 | 11.2 | 1 470 (59.5%) | 2 164 (29.0%) | 2 594 (34.7%) |
| dbo:Journalist | 1 505 (91.4%) | 2 336 (71.1%) | 2 522 (76.8%) | 7.3 | 9.2 | 12.0 | 1 075 (65.3%) | 1 324 (40.3%) | 1 621 (49.3%) |
| dbo:Presenter | 601 (93.5%) | 2 608 (55.8%) | 2 894 (61.9%) | 7.6 | 5.8 | 8.3 | 443 (68.9%) | 977 (20.9%) | 1 132 (24.2%) |
| dbo:Organization | 96 145 (56.2%) | 169 332 (31.9%) | 189 923 (35.8%) | 6.6 | 13.3 | 14.2 | 58 041 (33.9%) | 37 374 ( 7.0%) | 61 496 (11.6%) |
| dbo:Company | 27 691 (61.0%) | 50 778 (42.5%) | 55 429 (46.4%) | 6.4 | 12.0 | 12.7 | 19 288 (42.5%) | 12 972 (10.9%) | 21 639 (18.1%) |
| dbo:EducationalInst. | 12 659 (43.8%) | 13 515 (12.5%) | 18 197 (16.8%) | 4.6 | 5.7 | 5.8 | 6 375 (22.0%) | 2 366 ( 2.2%) | 6 467 ( 6.0%) |
| dbo:Group | 27 787 (87.0%) | 39 472 (64.1%) | 42 702 (69.3%) | 7.9 | 19.7 | 22.5 | 18 042 (56.5%) | 11 198 (18.2%) | 11 035 (17.9%) |
| dbo:SportsTeam | 9 888 (42.8%) | 18 767 (31.7%) | 20 680 (34.9%) | 5.5 | 11.5 | 11.1 | 5 218 (22.6%) | 2 067 ( 3.5%) | 7 464 (12.6%) |
| dbo:Broadcaster | 9 921 (75.4%) | 18 674 (53.3%) | 20 528 (58.6%) | 8.7 | 10.9 | 13.3 | 5 313 (40.4%) | 3 263 ( 9.3%) | 4 938 (14.1%) |
| dbo:MilitaryUnit | 1 934 (14.6%) | 1 754 ( 5.1%) | 2 221 ( 6.4%) | 4.7 | 10.0 | 9.3 | 753 ( 5.7%) | 144 ( 0.4%) | 391 ( 1.1%) |
| dbo:PoliticalParty | 1 019 (43.1%) | 2 804 (20.4%) | 3 370 (24.5%) | 6.2 | 15.6 | 14.3 | 422 (17.8%) | 430 ( 3.1%) | 666 ( 4.8%) |
| dbo:GovernmentAgency | 680 (41.1%) | 1 522 (17.6%) | 2 134 (24.7%) | 5.4 | 9.8 | 9.6 | 225 (13.6%) | 301 ( 3.5%) | 664 ( 7.7%) |
| dbo:Non-ProfitOrg. | 1 315 (55.9%) | 2 585 (33.3%) | 2 933 (37.7%) | 5.9 | 10.2 | 10.6 | 874 (37.1%) | 886 (11.4%) | 1 610 (20.7%) |
| dbo:TradeUnion | 51 (33.3%) | 862 (42.5%) | 760 (37.5%) | 5.2 | 18.7 | 16.2 | 24 (15.7%) | 80 ( 3.9%) | 164 ( 8.1%) |
| All entities | 620 396 (71.0%) | 906 349 (35.3%) | 1 026 413 (40.0%) | 6.9 | 12.7 | 13.7 | 304 773 (34.9%) | 271 824 (10.5%) | 322 124 (12.5%) |

## 4.4 Using SocialLink

As stated in Section 4.1, SocialLink establishes a link between DBpedia and Twitter, centered on popular entities occurring in both of them, which enables transferring knowledge from one resource to another and back, as well as comparing and jointly analysing the DBpedia graph and Twitter network. In the following, we describe four example use cases where these capabilities can be leveraged.

### 4.4.1 DBpedia to Twitter: User Profiling

The task of inferring users attributes based on their digital footprint is typically referred to as *user profiling*. Prediction of various attributes based on a person's social graph, posted content, or other attributes is popular among researchers and companies. However, in most setups, namely supervised machine learning-based ones, user profiling requires significant amounts of manual labour to construct training sets. This both limits the possible attributes that can be inferred and the applicability of approaches operating on large amounts of training data, such as DNNs. Recently, researchers focused on automatic crawling of user profiling datasets from social media. However, even the largest datasets only contain few thousands examples per property (Farseev et al., 2015) and are limited to properties explicitly present in social media.

SocialLink helps tackling user profiling by providing accurate machine-readable descriptions for hundreds of thousands of social media profiles. Any attribute present in DBpedia can now be modeled without relying on expensive manual annotation, and SocialLink can be used both to train and evaluate any proposed attribute classifiers.

Another example is inferring user interests based on social graph. Consider a user following, mentioning, or otherwise interacting with accounts aligned in SocialLink. By using this information, one can try to model interests, location, and language of the user by just looking at the DBpedia properties of these accounts (Besel et al., 2016; Piao and Breslin, 2017). For instance, following dbr:SpaceX and dbr:NASA can point on a dbr:Aerospace_engineering industry fan, while many dbr:Donald_Trump-related tweets can reveal a dbr:GOP supporter. The ability of SocialLink to significantly simplify user profiling pipelines is demonstrated in Chapter 6.

### 4.4.2 DBpedia to Twitter: Entity Linking

Another use case is the Named Entity Linking (NEL) task, whose goal is to link mentions of named entities in a text to their corresponding entities in a KB such as DBpedia. Challenging on its own, the NEL task presents additional unique challenges when applied

to social media posts due to noisiness, lack of sufficient textual context, and informal nature of posts (e.g., use of slang).

Social media posts typically contain explicit mentions of social media accounts in the form of @username snippets. When referring to Twitter, some of these mentions (especially the ones referring to popular accounts) may be aligned in SocialLink, and thus can be directly disambiguated to DBpedia with high precision using our resource. Apart being part of the NEL result, these links provide additional contextual information (injected from DBpedia) that can be leveraged for disambiguating other named entities occurring in the post being processed. Additionally, mentions of the named entities in social media posts are typically done via the specific @username constructs that further hinders the usage of established Natural Language Processing toolsets. SocialLink contain direct links between the social media profile (identified by the unique username) and corresponding DBpedia entities, which is the usual target of the NEL task. Therefore, disambiguating and linking a named entity could be as simple as a simple lookup in our resource. SocialLink was used in this capacity by two teams (Corcoglioniti et al., 2016; Minard et al., 2016) participating to a NEL challenge on Italian tweets (NEEL-IT task) as part of the EVALITA 2016 campaign, allowing both of them to improve their results.

It is worth noting that the two-step approach of the SocialLink pipeline can be adapted to directly disambiguate named entities in texts against the social media. Such functionality is present in the Social Media Toolkit which will be described in Chapter 7.

### 4.4.3 Twitter to DBpedia: Extracting FOAF Profiles and Type Prediction

Up-to-date information about DBpedia persons and organizations can be extracted from Twitter and brought to DBpedia after an alignment is established through SocialLink. Focusing on persons, different profile properties expressible with FOAF may be extracted from a DBpedia person's Twitter account, including:

- basic properties like foaf:name, foaf:surname, foaf:gender, foaf:birthday, and foaf:depiction linking to user images scarce in DBpedia but available in Twitter profiles;
- acquaintances (foaf:knows), extracted from friends, followers and Twitter accounts a user interacted with that are aligned to DBpedia entities in SocialLink;
- links to homepages (foaf:homepage and similar) and other web resources from a Twitter user description and posts, that can be matched to external links in DBpedia to mine relations with other DBpedia entities (e.g., affiliation, authorship, participation, all expressible in FOAF).

While a basic FOAF profile can be extracted from any Twitter account, the links to DBpedia provided by SocialLink allow grounding the extracted data and disambiguating

the values of object properties with respect to a larger KB, this way increasing the usefulness of extracted FOAF profiles.

Going further, as will be shown in Chapter 5, Twitter data imported along the populated links can be used as features to infer missing type information for entities in DBpedia using simple machine learning-based approaches. There we demonstrate that features based on Twitter data can outperform state-of-the-art entity representations built from a knowledge graph on a wide selection of DBpedia types. The performance is further improved by combining those feature families together.

Both of these use cases demonstrate the knowledge transfer from Twitter to DBpedia by showing that social media data can serve to directly enrich a target knowledge graph. Additionally, the results presented in Chapter 5 indicate possibility of using the same approach for ontology population and user profiling.

### 4.4.4 Twitter to Wikidata: Referencing of Crowdsourced Knowledge

Instead of bringing the knowledge directly, social media can be used as a source of external references for existing claims in datasets in the LOD. Nowadays, public crowdsourced datasets, such as Wikidata, are trying to find ways to corroborate the knowledge contained there with external sources. Currently, only a quarter of all claims contained in Wikidata is supported with external (non-wiki) references. Since many of the profiles in social media are verified and contain first-hand information about the entities they represent, links to them constitute in many cases valid external references. Given sufficient high-quality links between a crowdsourced dataset and social media, which SocialLink could provide, we can significantly improve the coverage of references in such dataset.

The ongoing `soweego`[13] project, recently funded by the Wikimedia Foundation, aims to use SocialLink approach to link external catalogs (including social media) to Wikidata. While SocialLink dataset already provides a significant number of links that could be used to provide references (we mostly use Wikidata URIs in the resource), a custom version explicitly targeting Wikidata can be produced using the same or improved pipeline.

## 4.5 Conclusions and Future Work

In this chapter, we presented our Linked Open Data dataset that links Twitter profiles to corresponding DBpedia entities in multiple language chapters. Building on the approach described in Chapter 3, we have made the SocialLink dataset a valuable resource for the Semantic Web community and Social Media researchers alike. Use cases of SocialLink include, but are not limited to, user profiling, entity linking, and knowledge base

[13] https://meta.wikimedia.org/wiki/Grants:Project/Hjfocs/soweego

enrichment. Our resource can be automatically repopulated using an open source software allowing reproducibility and welcoming contributions from the community. To this date, we have released three major revisions of the dataset, each marking significant milestones in the SocialLink development.

# Chapter 5

# Type Prediction Combining Linked Open Data and Social Media

In Chapter 3 and 4 we have presented the approach for linking representations in Linked Open Data (LOD) and social media of the same real-world entities, such as persons and organizations. Our SocialLink efforts enable knowledge transfer between the two worlds, making it possible to combine and leverage data in prediction problems, complementing the relative strengths of the two mediums: while the LOD knowledge is highly structured but also scarce and obsolete for some entities, social media data provide real-time updates and increased coverage, albeit being mostly unstructured.

In this chapter, we investigate the feasibility of using social media data to perform type prediction for entities in the LOD knowledge graph exploiting the above-mentioned knowledge transfer. We discuss how to gather training data for such a task, and how to build an efficient domain-independent vector representation of entities based on social media data. While in Chapter 3 we perform heavy feature engineering with the goal to achieve the best possible performance on the linking task, here we identify generic feature families to cover data that can be extracted from the stream of social media content to be used in multiple tasks. Our experiments on several type prediction tasks using DBpedia and Twitter data show the effectiveness of this representation, both alone and combined with knowledge graph-based features, suggesting its potential for ontology population. Results presented in this chapter prove that the ingestion of social media data to benefit tasks in LOD is possible once enough links are available. Results presented in this chapter were previously published in Nechaev et al. (2018c).

## 5.1 Introduction

The Linked Open Data (LOD) cloud over the years has become a prominent source of background knowledge for a large selection of tasks. DBpedia and Wikidata in particular

cover a wide range of entities including significant amounts of people and organizations. The data in such knowledge graphs is highly structured and readily available. However, it is mainly populated and updated using the crowdsourced content-editing efforts of the Wikimedia community, making the contained knowledge often incomplete, noisy and stale. While a significant amount of research has been done to address those issues via ontology population from external sources, reasoning and knowledge graph completion, they are far from being solved. Social media, on the other hand, provide up-to-date information on an overwhelming amount of people, organizations, and brands: Facebook alone has more than 2.1B monthly active users. This data is hard to extract due to API limitations and privacy considerations, and difficult to process due to its semistructured nature. Despite difficulties, social media is steadily becoming a primary source of real-time knowledge: where a community-curated knowledge graph can take hours to months to update, the information in social media often appear immediately.

There is a considerable overlap of coverage between LOD and social media. Living people and existing organizations from LOD are likely to be found in the social media as well. Such entities have become increasingly interlinked. More than 50K links between DBpedia entities and corresponding Twitter profiles can be found in DBpedia, with more available in Wikidata. Current community efforts, such as our SocialLink project (see Nechaev et al., 2017b,d, 2018b, Chapter 4), aim to expand the coverage of those links significantly with the latest version of the SocialLink dataset ($v3.0$) providing additional 322K alignments from DBpedia to Twitter even with the conservative thresholds. The increased interlinking can enable knowledge transfer between the highly structured LOD cloud and the vibrant social media world, improving and simplifying the processing pipelines in both.

In particular, the ontology population task can benefit from such interlinking. In this task, the aim is to fill missing connections in the knowledge graph and to populate it with new data from an external resource, to improve the coverage on a particular set of attributes. This is particularly important, for example, for DBpedia, where such a fundamental attribute of a person as age has only 52,8% coverage. Even if a particular attribute has a good enough coverage, the problem of referencing appears. In Wikidata, for instance, each claim can be corroborated by references to external resources confirming the data. Given that many social media profiles of people and organizations are marked as "verified" and considered official, they can serve as references for existing facts.[1]

In this chapter, we consider and investigate the feasibility of using social media data to predict entity attributes and perform ontology population leveraging the links between the two. To the best of our knowledge, no one has tried to tackle these tasks exploiting social

[1]See, e.g., the ongoing Soweego Wikimedia project: https://meta.wikimedia.org/wiki/Grants:Project/Hjfocs/soweego

media data before. To this end, we present a general purpose approach that, given a stream of social media data (Twitter) and some links from entities in a LOD knowledge graph (DBpedia) to their social media profiles, is able to predict and fill entity type attributes. This is achieved by treating the knowledge graph as a source of ⟨entity, type⟩ training data, and the social media as a source of features for the considered type prediction task, which we use to build rich, domain-general entity representations by integrating different data in the social media entity profile (e.g., profile attributes, social graph, textual data).

An approach like the presented one poses different challenges, which we tackle in this chapter. Firstly, social media data is hard to acquire at scale. The Twitter API, like any social media API, imposes limits on the number of queries per fixed period, which makes it impractical to extract several important feature families, such as the social graph. Instead, we rely on a sufficiently large random sample of tweets obtained from Twitter Streaming API (in our experiments we utilize four years worth of stream data as we did in our linking approach presented in Chapter 3). We design our features having this random sample in mind allowing us to successfully extract features for up to 291M Twitter users. For example, the social graph is approximated from retweet and mention relations extracted from the stream of tweets available. Secondly, social media data is sparse and noisy: for some users, we might have a clean and complete profile and a significant amount of textual content, while for some, we might only have mentions of the user without any authored content. To overcome this issue, the feature space efficiently combines different kinds of user-related data in a joint representation, aimed to be effective in different type prediction tasks.

To evaluate our approach, we compare the proposed feature space derived from Twitter data with a state-of-the-art entity representation model for DBpedia by Cochez et al. (2017b), consisting of RDF entity embeddings. Our experiments show that our social media entity representation gives prediction performances competitive with the ones obtained using RDF embeddings, outperforming it in most of the considered type prediction tasks. Additionally, we show that by combining social media and RDF embeddings, performances can be further improved. Finally, in the same setting, we compare the usage of social media data to Wikipedia, a traditional source of knowledge for the DBpedia population task. We demonstrate that the social media data is able to complement the Wikipedia-based features effectively achieving up to 92% $F_1$. Overall, we demonstrate the effectiveness of performing the knowledge transfer from Twitter to DBpedia for the type prediction task even in cases where additional resources, such as Wikipedia, may be utilized.

While in this chapter we experiment only with DBpedia and Twitter data, the presented approach is in principle social media and knowledge graph-agnostic. From the social media side, the same data as we use here (social graph, posts, minimal user profile) can be found in virtually any other social network. On the knowledge graph side, the distributional

semantics-based features we use for DBpedia can be produced from an arbitrary knowledge graph (Grover and Leskovec, 2016).

The rest of the chapter is organized as follows. In Section 5.2 we introduce the considered type prediction task and its application to ontology population in detail. Section 5.3 provides an overview of our approach, with Section 5.4 detailing the acquisition of ground truth data from LOD and Section 5.5 describing the use of social media features to represent entities. In Section 5.6 we evaluate our approach on several type prediction problems. Related work is presented in Section 5.7, while Section 5.8 concludes.

## 5.2 Problem Definition

In this work, we investigate the combined use of LOD and social media data for predicting entity attributes, and more specifically entity *types*. This task is often referred as *type prediction*, and is characterized by the fact that the types being predicted form a closed set whose size is typically small, whereas other attribute prediction tasks involve a large and possibly open set of predicted values, such as all the entities in a knowledge graph for *link prediction* tasks, or a continuous range of values for regression tasks.[2]

In our context, the types being predicted come from a LOD knowledge graph. Typically, these types are ontological classes *explicitly* defined in the graph, but they may be also *implicitly* derived from other kinds of information, such as age category types derived from a birth date property (e.g., young adult, see Section 5.4). In other words, we refer here to a generic notion of type that is compatible with different attribute prediction problems.

The type prediction task can be seen as *classification* task whose labels are types, and whose flavor depends on the relations existing among the predicted types:

- a *binary classification* task arises whenever the considered types are independent, and thus a yes/no prediction may be independently produced for each type;
- a *multi-class classification* task arises when the considered types are mutually disjoint, so that a given entity must be assigned exactly to one of the considered types;
- a *multi-label classification* task occurs when types are not all disjoint (so an entity may have multiple types, i.e., labels) and there might exist dependencies among types (e.g., selected disjoint or sub-type constraints) that make inappropriate the use of independent binary classifiers whose output may be inconsistent. When these dependencies come in the form of a type hierarchy (e.g., via rdfs:subClassOf relation), the task is also referred as *hierarchical multi-label classification* (Melo et al., 2016; Rico et al., 2018).

[2]Using social media data may be feasible also in these tasks but it is out-of-scope here.

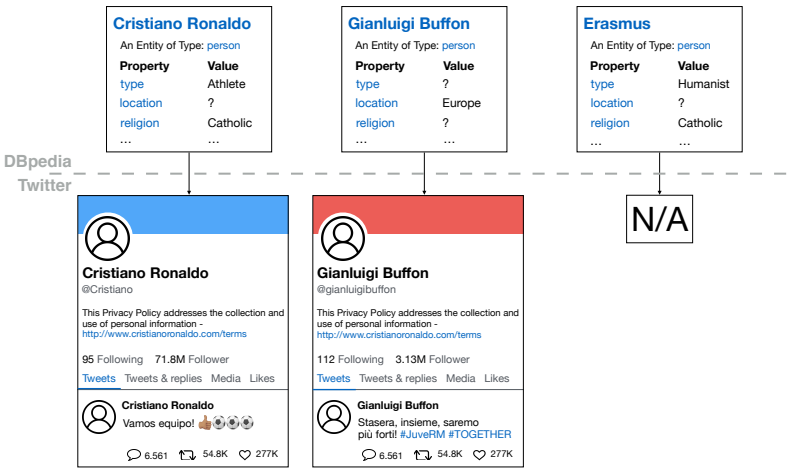



Figure 5.1. Using DBpedia-Twitter links for type prediction and ontology population.

In this work, we consider the case of mutually disjoint types and thus multi-class classification. This variant is general enough to be useful in practice, and permits us to focus on validating the feasibility of our approach without considering the additional complexity of (hierarchical) multi-label classification, which we leave as future work. In particular, by requiring an entity to belong to exactly one of a closed set of disjoint types, we rule out the problem of dealing with incomplete type knowledge (assuming an open-world stance), as the fact that an entity $e$ is not associated to type $t_i$ in the knowledge graph can be considered as a negative example for $t_i$ (instead of a case of missing information) if we know that $e$ has another type $t_j$ disjoint with $t_i$.

For predicting our types, we consider the use of social media data, either alone or combined with LOD data, to build the feature vector representations of predicted entities. We consider Twitter, but our approach and experiments can be in principle reproduced for other social networks. Social media data is obtained by aligning the LOD entity to its corresponding social media profile, from which different kinds of data can be extracted and mapped to features. A sizable amount of ⟨DBpedia entity, Twitter profile⟩ links are in DBpedia, and many more can be generated with adequate precision via SocialLink (see Chapter 4). As social media profiles are usually associated to living persons and organizations, these are the main kinds of entities for which our approach is applicable, and the only ones here considered for simplicity.

The main application we foresee for the type prediction task here investigated is *ontology population*, where a supervised type predictor is trained based on entities with known type information in LOD and then applied to classify and populate LOD entities without that information. This scenario is exemplified in Figure 5.1. It shows several DBpedia entities (top row) linked to their Twitter profiles (bottom row). Each of these links allows building a training example where the features are extracted from Twitter and

the type label is derived, explicitly or implicitly, from one of the entity's properties. In the example, the entity dbpedia:Cristiano_Ronaldo provides labeled examples to train classifiers for types dbo:Athlete (from property rdf:type) and dbo:Catholic (property dbo:religion). Then, these classifiers can be used to predict the rdf:type and dbo:religion types for the entity dbpedia:Gianluigi_Buffon. No examples can be extracted from the entity dbpedia:Erasmus as there is no link to Twitter.

The same supervised type predictor can be used for the *user profiling* task. Here, instead of predicting a type for an existing entity in the knowledge graph, an arbitrary user can be placed into the knowledge graph connected to the known entities from this user's follow list. Then, the predicted type (e.g., location, age) can be assigned to this user. This direction has been partially investigated in the relevant literature (Piao and Breslin, 2018) and is out of the scope of this chapter.

## 5.3 Approach Overview

The two main steps for building a supervised solution for type prediction are (i) the acquisition of the necessary *ground truth* ⟨entity, type⟩ pairs, and (ii) the construction of an effective feature vector *entity representation*. These two steps are detailed respectively in Sections 5.4 and 5.5. Together, they permit to build a training set out of which a supervised classifier can be trained and/or evaluated, as shown graphically in Figure 5.2.

In the figure, the presence of the social link between entity dbpedia:Cristiano_Ronaldo and his @Cristiano Twitter account allows us to build an entity representation based on the information in Twitter, with different kinds of Twitter data (e.g., profile data with description and location, content posted, and mentions and retweets received, social graph) encoded in different sub-spaces of the entity feature vector. The connection to other resources, e.g., RDF2Vec, can be used to further enrich the feature space. Finally, the type *Athlete* from the LOD entity description is used to label the example. Iterated over multiple ⟨entity, profile⟩ pairs where the predicted type(s) are known, this approach permits to train a supervised type predictor for other, possibly unseen ⟨entity, profile⟩ pairs where those types are not known and can thus be populated.

In deriving the entity representation, we avoid any fine tuning for a specific prediction task, and strive for capturing all the available information in a *single*, *comprehensive*, *domain-independent* representation that can be effective in different type prediction tasks. So, for instance, the same vector representation shown in Figure 5.2 can be used to predict other types, e.g., related to the missing location property, in which case the training examples are extracted from all entities that have a valid location.

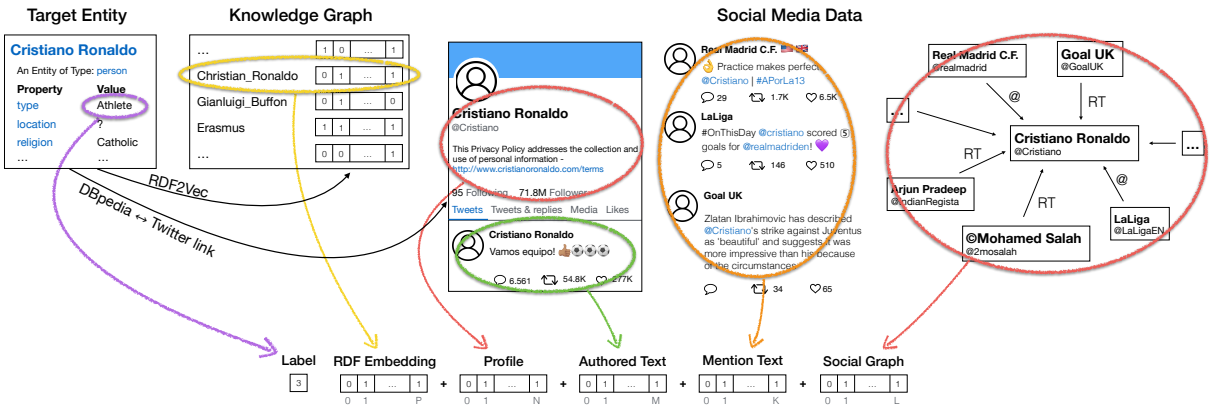



Figure 5.2. Example of training sample generation from DBpedia and Twitter data.

## 5.4 Ground Truth Acquisition from LOD

In the considered multi-class prediction scenario, the acquisition of ground truth data consists in extracting one or more properly-defined *type prediction tasks* out of a given LOD knowledge graph (e.g., DBpedia) aligned to the considered social network (e.g., Twitter), each task consisting of a set of disjoint predicted types $T = \{t_1 \dots t_n\}$ and a dataset of $\langle$entity, profile, type$\rangle$ triples $\langle e, p_e, t_e \rangle$, $t_e \in T$ that is amenable to training a supervised classifier. Here, we describe the ground truth acquisition methodology that we manually applied to extract 8 type prediction tasks out of DBpedia for our experiments, leaving the definition of an automated procedure for ground truth acquisition (possibly based on the methodology) as future work.

### 5.4.1 Methodology

We followed the three-step methodology described next.

**Step 1: Type Information Identification** This step deals with identifying the sources of type information available in the considered knowledge graph, leveraging TBox information and data querying to extract necessary ABox statistics. In a knowledge graph, type-related information is available either explicitly or implicitly. Explicit type information comes in the form of rdf:type ABox triples assigning entities to well-known ontological classes. Implicit type information can be derived from other ABox triples involving the entity, in different ways. For instance:

- the object of the triple may already represent a type, as happens with values dbpedia:Male and dbpedia:Female of DBpedia property dbo:gender;
- the object of the triple may be part of a hierarchy whose upper nodes can be seen as types, as happens, e.g., with locations that can be spatially aggregated, or political parties that can be aggregated based on international affiliation;

- the object of the triple may be a numeric value or a date that can be discretized to different value ranges, an example being property dbo:birthDate that can be mapped to different age categories (e.g., child, young adult).

All the cases above can be treated as the materialization of implicit rdf:type triples via a rule-based strategy (e.g., via logical inference). These triples (if needed, see next steps) may be produced in a pre-processing step, so from now on we consider type information as fully available via rdf:type triples without loss of generality.

**Step 2: Prediction Task Selection** This step deals with the selection (here done manually based on the requirements below) of a set of prediction tasks based on the identified type information, each task defined by its set of types $T$. In our multi-class prediction context, the selection must satisfy two requirements:

- types $t_i \in T$ should be mutually disjoint; if not the case, overlapping types may be aggregated along a hierarchy or manually by adding super-classes, or alternatively finer-grained types representing their intersections may be introduced;

- for each type $t_i \in T$, there should be enough samples $\langle e, p_e, t_i \rangle$ for training a classifier; if it is not the case, types $t_i$ may be aggregated along a hierarchy or manually to obtain coarser-grained types satisfying this requirement.

While these requirements are enough for our experiments, in a real ontology population scenario the selection should also satisfy two additional requirements:

- for each entity $e$ in the populated knowledge graph and for each prediction task with types $T$, it must be possible to decide whether $e$ may accept a type $t_i \in T$ (and thus the classifier may run on $e$). E.g., before predicting the age category of $e$, one should determine that $e$ is a person. This check may be formulated based on coarser-grained entity types associated to $e$ (e.g., a person/organization classification), either already known or predicted via an upstream classifier. Alternatively, the problem can be framed as a hierarchical type prediction task, where we can naturally predict correct sub-types based on a chosen parent type;

- for a prediction task to be useful, in addition to a large enough training set there must be a large enough amount of entities for which the predicted type information is missing and can thus be populated by the trained predictor.

**Step 3: Dataset Extraction** This step deals with extracting the $\langle e, p_e, t_e \rangle$ dataset for each selected prediction task. This involves implementing the pre-processing of Step 1 (if any) and the aggregations of Step 2 (if any). This step may involve also some *data*

*normalization* — e.g., to convert organization revenues to the same currency, for predicting a revenue class — and *data clean-up* — e.g., to discard wrong data such as football teams being assigned a dbo:gender, or discard entities associated to multiple incompatible types, such as a person with multiple age categories.

### 5.4.2 DBpedia Type Prediction Tasks

We applied the aforementioned methodology to extract a non-exhaustive set of 8 type prediction tasks for our experiments (see Section 5.6) from the living persons and organizations in DBpedia 2016-04 that are aligned to Twitter in DBpedia (via property dbo:wikiPageExternalLink). Detailed information, code, datasets (including ground truth data) and additional experiments are available online.[3] We briefly describe these tasks below:

- *Category* — this task covers the specific category of person (e.g., artist, athlete) or organization (e.g., company, government agency) using a set of 17 types corresponding to DBpedia classes that can be reasonably considered as disjoint. Special types for "other person" and "other organization" were added to aggregate persons and organizations belonging to types without enough training examples;
- *Location* — this task classifies entities (both persons and organizations) based on their location (property dul:hasLocation), aggregated geographically (property geonames:parentFeature) to obtain a 6-type continent-level classification;
- *Political Party* — this task classifies person entities based on political party (property dbo:party), aggregated along their affiliation to international party federations (property dbo:internationalAffiliation) to obtain a 6-type classification;
- *Religion* — this task classifies persons based on religion (property dbo:religion), manually aggregated in a 5-type classification (e.g., by merging different Christian divisions);
- *Age* — this task classifies person in 6 age categories (e.g., young adult 25-34 years old), computed based on their birth dates (property dbo:birthDate);
- *Org. Size* — this task classifies organizations based on their numbers of employees (property dbo:numberOfEmployees), which is discretized in a 4-type classification;
- *Revenue* — this task classifies organizations based on their revenue (property dbo:revenue), which is normalized to use US dollars as currency, cleaned-up discarding

[3] http://sociallink.futuro.media/type-prediction

Table 5.1. DBpedia type prediction tasks, with entity parent type (for the task being applicable), # of predicted types, and # of entities w/ type (training set) and w/o type (population target) in the DBpedia fragment linked to Twitter.

| Task | Parent type | Predicted types | # Samples (training) | # Samples (population) |
|---|---|---|---|---|
| *Category* | owl:Thing | 17 | 49,639 | n/a |
| *Location* | owl:Thing | 6 | 38,153 | 14,134 |
| *Political Party* | dbo:Person | 6 | 1,912 | 37,143 |
| *Religion* | dbo:Person | 5 | 1,858 | 37,197 |
| *Age* | dbo:Person | 6 | 31,998 | 6867 |
| *Org. Size* | dbo:Organisation | 4 | 1,062 | 12,171 |
| *Revenue* | dbo:Organisation | 3 | 412 | 12,821 |
| *Music Skill* | dbo:MusicalArtist | 3 | 7,085 | 277 |

outlier values (e.g., organizations with only few hundreds dollars revenue) and then discretized to form a 3-type classification;

- *Music Skill* — this task classifies musical artists based on their specialty (singers, instrumentalists, other), on the basis of DBpedia datatype property dbo:background.

Table 5.1 provides relevant statistics for the 8 prediction tasks extracted. For each task, it reports the number of predicted types (i.e., classes) and the number of entities in the fragment of DBpedia linked to Twitter here considered for which the type information to predict is respectively available — in which case a training sample is obtained — or missing — in which case the entity becomes a target for ontology population. The table also shows the *parent type* (with respect to predicted types) that entities must have for a prediction task being applicable (e.g., the *Political Party* task applies only to persons). Due to ongoing efforts in the Semantic Web community in interlinking LOD entities and social media profiles, including our own as detailed in Chapter 4, we expect a significant increase in both the number of training samples and potential ontology population targets (see Table 5.5).

## 5.5 Entity Representation with Social Features

As shown in Figure 5.2, we build an entity representation starting from social media data linked to the entity, possibly augmented with RDF features. As RDF features consist of existing RDF embeddings, our focus and contribution here is on the extraction of *social* features from the social network we consider in this work, i.e., Twitter.

We start by obtaining the list of Twitter accounts we are interested in by following the links from DBpedia to Twitter. Then we process the 4 TB of tweets gathered covering the period from 2013 until 2017 using the Streaming API, filtering out the tweets not

Table 5.2. Coverage (i.e., percentage of entities having the feature) and dimensionality statistics for the social features extracted from Twitter. Differences in coverage stem from information unavailability in the Twitter stream.

| Source | Feature | Coverage | # Dimensions |
|---|---|---|---|
| Text of tweets authored by the user | Text (LSA) | 79.5% | 100 |
| | Text (Bag-of-words) | 79.5% | 972,001 |
| | Hashtags (Bag-of-words) | 62.9% | 169,679 |
| Text of tweets mentioning the user | Text (LSA) | 86.1% | 100 |
| | Text (Bag-of-words) | 86.1% | 972,001 |
| Profile | Language | 79.5% | 47 |
| | Top-level domain of a URL | 79.5% | 288 |
| | Followers count | 79.5% | 1 |
| | Friends count | 79.5% | 1 |
| | Listed count | 79.5% | 1 |
| | Favorites count | 79.5% | 1 |
| | Statuses count | 79.5% | 1 |
| | Is protected | 79.5% | 1 |
| | Is verified | 79.5% | 1 |
| | Geo tagging enabled | 79.5% | 1 |
| | Has profile images/tiling | 79.5% | 4 |
| | Description (LSA) | 79.5% | 100 |
| Social graph | RT+mentions (sparse) | 87.9% | 2,203,062 |
| | RT+mentions (dense) | 100.0% | 300 |

related to users in the list. From the remaining tweets, we extract features based on four feature families: (i) social graph features; (ii) profile features; (iii) textual features from a user's own tweets; and (iv) textual features from tweets that mention the target user. Table 5.2 provides summary statistics for all the four feature families, which are detailed in the remainder of the section. The feature families we chose provide a comprehensive representation of available user information and performed well in various social media analysis tasks before (Zheleva and Getoor, 2009; Nechaev et al., 2017a; Li et al., 2014; Nechaev et al., 2018b). While we have already exploited some user-based features in Chapter 3, many of the features there are specifically designed to acquire similarity scores based on the comparison with the corresponding entity-based data. Instead, here the goal is to acquire a general representation of the user based on the available stream of tweets to be used in multiple unrelated tasks. For feature subspaces with increased sparsity, such as text and social graph, we also employ low-dimensional dense representations, i.e., embeddings.

**Textual Features** As seen in many studies tackling the user profiling task (see Section 5.7), the user-generated textual content can be a prominent feature exhibiting outstanding inference performance for attributes such as age, location, nationality, interests, and many others. We consider two sources of textual content for each user: the text that

is authored by the user and the text of tweets that are mentioning the target user. Text from both sources is accumulated and tokenized. Then special entities in the text, such as URLs, hashtags, and mentions, are filtered out. The resulting term sets are converted to sparse bag-of-words representations, with each term weighted using the tf-idf scheme. As with the social graph, we emit a dense embedding of each sparse vector. To this end, we employ Latent Semantic Analysis (LSA) to map a sparse vector into a dense one of size 100. In addition to text, we build a simplified bag-of-words representation considering only the hashtags that were used in tweets authored by the user.

**Profile-based Features** Each tweet object contains a snapshot[4] of profile data for the author of the tweet. To produce a representation based on this data, we collect the latest snapshot for each user and extract a variety of features from it. Among those are categorical features, such as a top-level domain name of the URL field and the user's self-declared language encoded as one-hot vectors, binary features for each boolean attribute in the profile, and the dense LSA representation of the user-supplied description. In total, the user object is converted into a feature vector of size 447.

**Social Graph Features** The social graph, while being regarded in the literature as one of the most prominent features for a variety of tasks including user profiling, community detection, and many others, is incredibly hard to obtain at scale. At the time of writing, Twitter allows the gathering of only 5,000 edges per minute via its REST API, which, in our case, translates to months of crawling time. Instead, we follow the procedure we introduced in Section 3.4.1. There the data extracted from the Twitter Streaming API is used to build an approximation of the real social graph. The approximation is computed by extracting mention and retweet interactions between users, where a "follow" edge is generated from the mentioning/retweeting user to the target one for each such interaction, yielding a graph with 2.7B edges. We introduce the resulting social graph into our feature model in sparse form with 2.2M dimensions, encoding each node as a bag of adjacent nodes with the appropriate weight. As in Section 3.4.1, we also introduce a dense representation for each user. To this end, we learn embeddings of size 300 for the 500K most frequently followed users by building the co-occurrence matrix and estimating factorization using the Swivel algorithm (Shazeer et al., 2016). This algorithm is inspired by the distributional semantics hypothesis for natural language, in that the users that interact with the same profiles will end up being similar in the resulting vector space. Then, to obtain the dense representation for an arbitrary user, we compute a weighted average of the embeddings of his/her friends in the top 500K list. If no friend has an embedding or if the user is not in the social graph, we emit a default representation by

[4]Twitter documentation article about the user object: https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/user-object

averaging all the available embeddings. This procedure provides perfect coverage allowing to produce a rough approximation even for users not seen in the Twitter stream.

## 5.6 Experiments

In this section, we investigate the feasibility of performing type prediction using social media data. Namely, we assess the use of Twitter-based features (as described in Section 5.5) alone and in conjunction with state-of-the-art DBpedia entity representations (RDF embeddings). Additionally, we compare those features with the ones derived from Wikipedia text as proposed in Aprosio et al. (2013). We conclude this section by providing an analysis of the contribution of dense social features to the overall performance.

### 5.6.1 Experimental Setting

We consider the type prediction tasks introduced in Section 5.4 and summarized in Table 5.1, each task consisting of a set of mutually disjoint types (e.g., different person and organisation categories) and a dataset of entities from DBpedia (version 2016-04) having those types and a link to a corresponding Twitter profile.

For each task, we consider the following prediction approaches:

- *MF* — a most frequent baseline always predicting the type with the largest number of entities in the task dataset, whose performances represent a lower bound of the performances achievable on the task, and give an idea of its difficulty;
- *RDF* — a linear Support Vector Machine (SVM)[5] classifier using the *PageRank Split* RDF embeddings for DBpedia 2016-04 (Cochez et al., 2017b) as entity representations. This particular embedding weighting schema was selected based on preliminary experiments, the details of which are available online;[6]
- *Social* — a linear SVM using our Twitter-based social features introduced in Section 5.5 as entity representations, produced for all the task entities based on the tweets sampled in 4 years from the Twitter Streaming API;
- *Social+RDF* — a linear SVM using a combination of social features and RDF embeddings as entity representations.

It is worth pointing out that the *RDF* embeddings we used[7] were computed from a knowledge graph that includes the entity types being predicted for the tasks *Cate-*

[5]We use the LibLinear (Fan et al., 2008) software package with L2 regularization and L1 (Hinge) loss. LibLinear was chosen as it copes well with large feature spaces. We tested other classifiers (SVM with Gaussian/polynomial kernels, Random Forests) obtaining similar performances but much higher run times.

[6]http://sociallink.futuro.media/type-prediction

[7]RDF embeddings downloaded from http://data.dws.informatik.uni-mannheim.de/rdf2vec/models/DBpedia/2016-04/GlobalVectors/

Table 5.3. Type prediction performances. Statistically significant differences w.r.t. *Social* are marked with $^+$ if better, $^-$ if worse; possibly overestimated performances (see text) are marked with $^*$.

| Task | Approach | $P_{macro}$ | $R_{macro}$ | $F_{1\,macro}$ | $P_{micro}$ | $R_{micro}$ | $F_{1\,micro}$ |
|---|---|---|---|---|---|---|---|
| Category | *MF* | $0.016^-$ | $0.059^-$ | $0.026^-$ | $0.277^-$ | $0.277^-$ | $0.277^-$ |
| | *RDF** | $0.612^-$ | $0.520^+$ | $\mathbf{0.539}^+$ | $0.759^+$ | $0.721^+$ | $\mathbf{0.740}^+$ |
| | *Social* | 0.666 | 0.387 | 0.445 | 0.667 | 0.610 | 0.637 |
| Location | *MF* | $0.069^-$ | $0.167^-$ | $0.097^-$ | $0.412^-$ | $0.412^-$ | $0.412^-$ |
| | *RDF** | $0.466^-$ | $0.299^-$ | $0.292^-$ | $0.592^-$ | $0.580^-$ | $0.586^-$ |
| | *Social* | 0.904 | 0.680 | **0.763** | 0.878 | 0.796 | **0.835** |
| Political Party | *MF* | $0.072^-$ | $0.167^-$ | $0.100^-$ | $0.431^-$ | $0.431^-$ | $0.431^-$ |
| | *RDF** | $0.441^-$ | $0.316^-$ | $0.327^-$ | $0.584^-$ | $0.541^-$ | $0.561^-$ |
| | *Social* | 0.721 | 0.527 | **0.596** | 0.812 | 0.728 | **0.767** |
| Religion | *MF* | $0.122^-$ | $0.200^-$ | $0.152^-$ | $0.610^-$ | $0.610^-$ | $0.610^-$ |
| | *RDF** | 0.723 | $0.358^-$ | $0.374^-$ | $0.688^-$ | $0.681^-$ | $0.684^-$ |
| | *Social* | 0.651 | 0.474 | **0.488** | 0.726 | 0.721 | **0.723** |
| Age Category | *MF* | $0.054^-$ | $0.167^-$ | $0.082^-$ | $0.326^-$ | $0.326^-$ | $0.326^-$ |
| | *RDF* | $0.266^-$ | $0.239^-$ | $0.222^-$ | $0.362^-$ | $0.362^-$ | $0.362^-$ |
| | *Social* | 0.423 | 0.320 | 0.325 | 0.451 | 0.450 | 0.450 |
| | *Social+RDF* | 0.431 | $0.332^+$ | $\mathbf{0.339}^+$ | $0.462^+$ | $0.456^+$ | $\mathbf{0.459}^+$ |
| Org. Size | *MF* | $0.076^-$ | $0.250^-$ | $0.117^-$ | $0.304^-$ | $0.304^-$ | $0.304^-$ |
| | *RDF* | $0.359^-$ | $0.379^-$ | $0.350^-$ | $0.401^-$ | $0.400^-$ | $0.401^-$ |
| | *Social* | 0.417 | 0.428 | 0.417 | 0.441 | 0.440 | 0.440 |
| | *Social+RDF* | $0.453^+$ | $0.458^+$ | $\mathbf{0.447}^+$ | $0.481^+$ | $0.477^+$ | $\mathbf{0.479}^+$ |
| Revenue | *MF* | $0.127^-$ | $0.333^-$ | $0.184^-$ | $0.381^-$ | $0.381^-$ | $0.381^-$ |
| | *RDF* | 0.514 | 0.509 | 0.480 | 0.522 | 0.515 | 0.518 |
| | *Social* | 0.555 | 0.533 | 0.531 | 0.548 | 0.544 | 0.546 |
| | *Social+RDF* | 0.539 | 0.530 | 0.519 | 0.541 | 0.539 | 0.540 |
| Music Skill | *MF* | $0.254^-$ | $0.333^-$ | $0.289^-$ | $0.763^-$ | $0.763^-$ | $0.763^-$ |
| | *RDF* | $0.255^-$ | $0.333^-$ | $0.289^-$ | $0.765^-$ | $0.762^-$ | $0.763^-$ |
| | *Social* | 0.664 | 0.424 | 0.436 | 0.804 | 0.792 | 0.798 |
| | *Social+RDF* | 0.653 | $0.434^+$ | $\mathbf{0.451}^+$ | 0.803 | 0.792 | 0.798 |

*gory*, *Location*, *Political Party*, and *Religion*. Therefore, the performances of approach *RDF* in these tasks may be overestimated, influencing the comparison with other approaches. This problem does not affect the remaining datasets, which were generated starting from datatype properties not used for producing the RDF embeddings. The RDF embeddings (Ristoski and Paulheim, 2016a) are used in many tasks in the literature outperforming other graph-based entity representations, both sparse and dense, therefore we employ them in our experiments as a baseline.

For each task dataset, we evaluate the performances of *RDF*, *Social* and *Social+RDF* classifiers via a 5-fold cross-validation protocol, where we collect predictions for all the entities in the task dataset by iteratively selecting one of the five partitions and applying on it the classifier trained on the remaining four partitions. During each training step (five in total), we apply a nested 3-fold cross-validation loop to select the optimal classifier

hyper-parameters (the regularization parameter $C$) and the classifier score threshold below which we should abstain in order to balance precision and recall (i.e., optimize $F_{1\,micro}$). This threshold is then used at prediction time to abstain and discard the types predicted with low confidence. The *MF* baseline never abstains.

As evaluation measures we use Precision (P), Recall (R), and F1 scores in their micro-averaged ($P_{micro}$, $R_{micro}$, $F_{1\,micro}$) and macro-averaged ($P_{macro}$, $R_{macro}$, $F_{1\,macro}$) variants, as commonly defined for multi-class classification problems.[8] We test the statistical significance of the difference of those scores via the *approximate randomization* test (Noreen, 1989) (significant if $p\text{-}value \leq 0.05$), and produce precision-recall curves by varying the abstain threshold on the prediction score returned by the classifier (the SVM margin). Both evaluation scores and statistical significance are computed on the predictions for the whole task dataset obtained via cross-validation.

The whole pipeline required a couple of days to extract features from the stream of tweets (using Apache Flink[9] ), 14 hours of GPU time to produce dense *Social* features, and additional 7 hours for the nested cross-validation training and evaluation of all the classifiers on the 8 type prediction tasks (10-core E5-2630 machine with 192 GB RAM and GeForce GTX 1080 GPU).

### 5.6.2 Experimental Results

Table 5.3 reports the performance scores obtained by the different approaches on the 8 type prediction tasks, with indication of statistically significant differences with respect to *Social* ($^+$ if significantly better, $^-$ if significantly worse). The first four tasks — *Category*, *Location*, *Political Party*, *Religion* — are the ones where predicted types were included in the RDF embeddings of approach *RDF*, whose performances may be overestimated (marked with $^*$); for these tasks, we do not consider the *Social+RDF* combination approach.

Compared to the baseline approach *MF*, *Social* always results in better performance (whereas approach *RDF* performs like the baseline in task *Music Skill*), with statistically significant differences that are higher for tasks *Location*, *Category*, and *Political Party*, somehow suggesting that these tasks are "easier" than the others.

[8] Given a prediction task with types $t_i \in T$, we define $tp_i$ (true positives), $fp_i$ (false positives), and $fn_i$ (false negatives) as the numbers of entities respectively: of type $t_i$ correctly classified as $t_i$; of type $t_j \neq t_i$ wrongly classified as $t_i$; of type $t_i$ wrongly classified as $t_j \neq t_i$. Based on that, we have:

$$P_{micro} = \frac{\sum_i tp_i}{\sum_i tp_i + \sum_i fp_i} \quad R_{micro} = \frac{\sum_i tp_i}{\sum_i tp_i + \sum_i fn_i} \quad F_{1\,micro} = \frac{2 \cdot \sum_i tp_i}{2 \cdot \sum_i tp_i + \sum_i fp_i + \sum_i fn_i}$$

$$P_{macro} = \sum_i \frac{tp_i}{tp_i + fp_i} \quad R_{macro} = \sum_i \frac{tp_i}{tp_i + fn_i} \quad F_{1\,macro} = \sum_i \frac{2 \cdot tp_i}{2 \cdot tp_i + fp_i + fn_i}$$

[9] https://flink.apache.org/

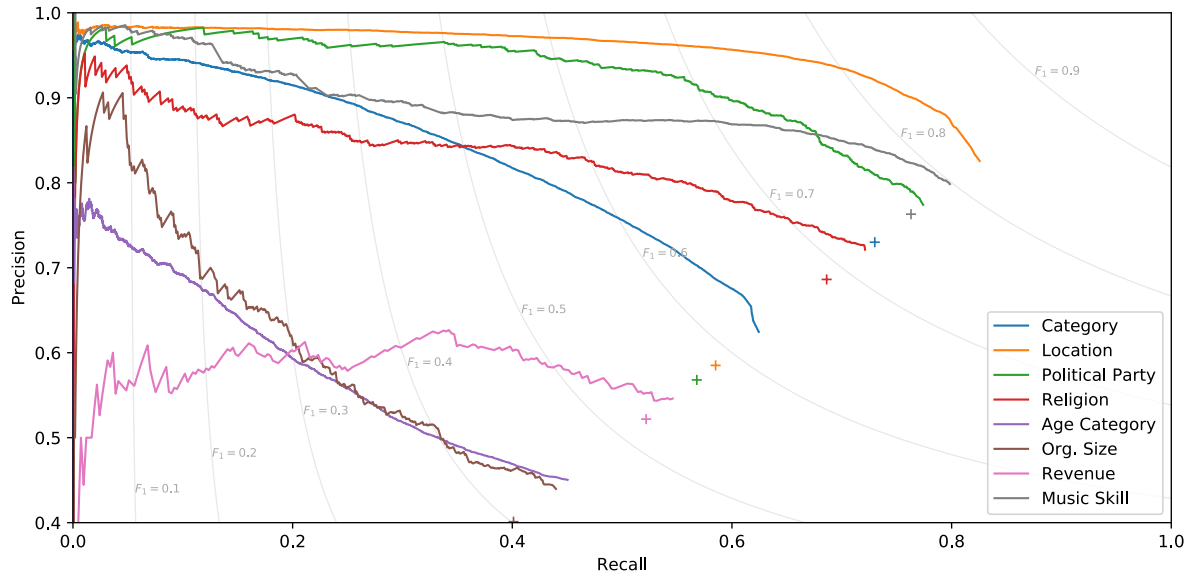



Figure 5.3. Precision-recall curves for different prediction tasks: lines correspond to approach *Social*, cross markers to approach *RDF* (best-$F_{1\,micro}$ setting).

Compared to approach *RDF*, *Social* outperforms *RDF* in all the 4 tasks where a proper comparison is possible (*Age*, *Org. Size*, *Revenue*, *Music Skill*), with statistically significant differences in 3 out of 4 tasks. In the other tasks, *Social* still manages to outperform *RDF* in 3 out of 4 tasks (*Location*, *Political Party*, *Religion*, with only exception of *Religion* $P_{macro}$), a remarkable result given that *RDF* performances may be overestimated in these tasks. We note that approach *RDF* significantly outperforms *Social* in task *Category*, whose data was obtained directly from DBpedia rdf:type triples.

The fact that approach *Social* is competitive with respect to approach *RDF* (outperforming it in most cases), suggests that the proposed social media entity representation may contribute positively in an ontology population task. This is confirmed by considering approach *Social+RDF* that combines *Social* and *RDF*. In the four tasks where we evaluate approach *Social+RDF*, it always outperforms *RDF* (expected, as *Social* outperforms it too on these tasks), and in 2 tasks out of 4 it also outperforms *Social* (score differences statistically significant except for $P_{macro}$ on task *Age*), showing that the combined representation is generally better than its components taken separately.

As the precision scores in Table 5.3 (corresponding to the best $F_{1\,micro}$ score) may not be satisfactory in an ontology population task, we investigate whether better precision scores can be obtained at the expenses of recall, by changing the threshold on the classifier score to vary the precision/recall balance. The results are reported in Figure 5.3, which plots the precision/recall curves ($P_{micro}/R_{micro}$) for approach *Social* on the different tasks (performances of approach *RDF* for best-$F_{1\,micro}$ setting plotted with cross markers for

Table 5.4. Type prediction performances in comparison and in conjunction with Wikipedia-based features — Aprosio et al. (2013). Statistical significance is shown with respect to *All*.

| Task | Approach | $P_{macro}$ | $R_{macro}$ | $F_{1\,macro}$ | $P_{micro}$ | $R_{micro}$ | $F_{1\,micro}$ |
|---|---|---|---|---|---|---|---|
| Category | *Social+RDF* | $0.784^{-}$ | $0.611^{-}$ | $0.658^{-}$ | $0.814^{-}$ | $0.797^{-}$ | $0.805^{-}$ |
| | Aprosio et al. (2013) | $0.888^{+}$ | $0.597^{-}$ | $0.704^{-}$ | $0.896^{+}$ | $0.667^{-}$ | $0.764^{-}$ |
| | *All* | 0.876 | 0.767 | **0.811** | 0.883 | 0.872 | **0.878** |
| Location | *Social+RDF* | $0.893^{-}$ | $0.717^{-}$ | $0.781^{-}$ | $0.865^{-}$ | $0.843^{-}$ | $0.854^{-}$ |
| | Aprosio et al. (2013) | $0.955^{+}$ | $0.674^{-}$ | $0.782^{-}$ | $0.961^{+}$ | $0.755^{-}$ | $0.846^{-}$ |
| | *All* | 0.934 | 0.840 | **0.880** | 0.931 | 0.923 | **0.927** |
| Political Party | *Social+RDF* | $0.729^{-}$ | $0.574^{-}$ | $0.631^{-}$ | $0.805^{-}$ | $0.757^{-}$ | $0.780^{-}$ |
| | Aprosio et al. (2013) | 0.798 | 0.587 | 0.657 | $0.815^{-}$ | $0.775^{-}$ | $0.795^{-}$ |
| | *All* | 0.801 | 0.621 | 0.687 | 0.871 | 0.807 | **0.838** |
| Religion | *Social+RDF* | $0.639^{-}$ | $0.536^{-}$ | $0.561^{-}$ | $0.752^{-}$ | $0.746^{-}$ | $0.749^{-}$ |
| | Aprosio et al. (2013) | 0.758 | $0.492^{-}$ | $0.544^{-}$ | $0.747^{-}$ | $0.737^{-}$ | $0.742^{-}$ |
| | *All* | 0.758 | 0.569 | **0.601** | 0.791 | 0.771 | **0.781** |
| Age Category | *Social+RDF* | $0.416^{-}$ | $0.320^{-}$ | $0.320^{-}$ | $0.455^{-}$ | $0.452^{-}$ | $0.453^{-}$ |
| | Aprosio et al. (2013) | $0.460^{-}$ | $0.382^{-}$ | $0.391^{-}$ | $0.498^{-}$ | $0.494^{-}$ | $0.496^{-}$ |
| | *All* | 0.488 | 0.421 | **0.434** | 0.526 | 0.523 | **0.525** |
| Org. Size | *Social+RDF* | $0.425^{-}$ | $0.438^{-}$ | $0.425^{-}$ | $0.456^{-}$ | $0.455^{-}$ | $0.455^{-}$ |
| | Aprosio et al. (2013) | 0.497 | 0.512 | 0.490 | 0.518 | 0.518 | 0.518 |
| | *All* | 0.509 | 0.494 | 0.496 | 0.531 | 0.505 | 0.518 |
| Revenue | *Social+RDF* | 0.571 | $0.536^{-}$ | $0.537^{-}$ | $0.564^{-}$ | $0.549^{-}$ | $0.556^{-}$ |
| | Aprosio et al. (2013) | 0.558 | 0.554 | 0.548 | 0.575 | 0.566 | 0.570 |
| | *All* | 0.606 | 0.584 | 0.586 | 0.608 | 0.595 | 0.601 |
| Music Skill | *Social+RDF* | $0.617^{-}$ | $0.427^{-}$ | $0.447^{-}$ | $0.806^{-}$ | $0.784^{-}$ | $0.795^{-}$ |
| | Aprosio et al. (2013) | 0.731 | 0.596 | 0.638 | 0.854 | 0.846 | 0.850 |
| | *All* | 0.736 | 0.598 | 0.638 | 0.851 | 0.844 | 0.848 |

reference). The plot shows that high precision levels ($P_{micro} > 0.9$) can be obtained for tasks *Category*, *Location*, and *Political Party*, thus opening up the possibility of performing ontology population from social media for these types.

### 5.6.3 Comparison to Wikipedia-based features

In addition to RDF embeddings, we investigate the usage of social media data along with Wikipedia-based features — a popular source of knowledge for ontology population. While DBpedia itself is primarily populated from Wikipedia infoboxes, the rest of the text on the corresponding Wikipedia page can still provide additional data for a given entity. Moreover, Wikipedia pages are typically available in many languages, allowing cross-language models to further boost the performance compared to the single language ones. We follow Aprosio et al. (2013) approach for extracting features from Wikipedia articles to perform type prediction on our 8 tasks.

We thus add two additional approaches to the comparison:

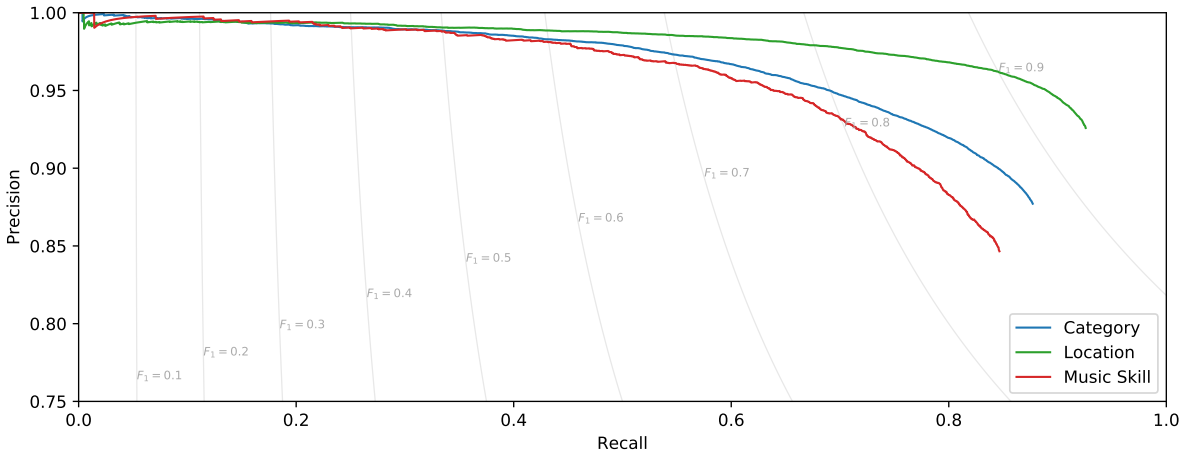



Figure 5.4. Precision-recall curves for top three performant prediction tasks using the approach *All*.

- Aprosio et al. (2013) — a linear SVM using the $K_{\mathbf{combo}}$ features acquired from Wikipedia as described by Aprosio et al. (2013). This includes Wikipedia categories, sections, templates and words (both LSA-based dense and sparse) extracted from six language variants of each article: English, Italian, German, French, Spanish, and Portuguese;
- *All* — a linear SVM using a combination of Aprosio et al. (2013), *Social* and *RDF* feature sets;

Table 5.4 reports the performances of these approaches when compared to the best performing models from Section 5.6.2: *Social+RDF*. The Aprosio et al. (2013) approach generally outperforms Social+RDF features (*Political Party*, *Age*, *Org. Size*, *Revenue* and *Music Skill*), while providing similar performance for the rest. The *All* approach achieve the best performances in all tasks, significantly better in five of them. Most notably, in *Location* task, this approach was able to reach 92% $F_{1\,micro}$, 7% higher than the next best model. Corresponding precision/recall curves showcasing the results of the top three performing tasks for *All* are displayed in Figure 5.4. Such results show that, given that the coverage may increase in the future, social media data is versatile enough to complement a variety of data sources for the ontology population task. Indeed, even with a very ambitious precision goal of 98% those systems would deliver from 20% to 40% recall potentially allowing significant amount of entities to be populated.

### 5.6.4 Dense Social Representations

When processing, analyzing and potentially releasing social media data, it is essential to consider the privacy aspect of such actions. Even though we used public data to build our entity representation, in case we ever release it, there is always a risk of reverse engineering

Table 5.5. Number of samples for each task when using data from SocialLink v2 compared to DBpedia (see Table 5.1).

| Task | # Samples (training) | | # Samples (population) | |
|---|---|---|---|---|
| | DBpedia | SocialLink | DBpedia | SocialLink |
| *Category* | 49,639 | 498,842 | n/a | n/a |
| *Location* | 38,153 | 330,803 | 14,134 | 168,039 |
| *Political Party* | 1,912 | 16,941 | 37,143 | 380,232 |
| *Religion* | 1,858 | 10,587 | 37,197 | 386,586 |
| *Age* | 31,998 | 309,018 | 6,867 | 87,070 |
| *Org. Size* | 1,062 | 10,871 | 12,171 | 90,798 |
| *Revenue* | 412 | 5,452 | 12,821 | 96,217 |
| *Music Skill* | 7,085 | 25,011 | 277 | 1,050 |

of our sparse textual and social graph-based subspaces and profile features, which may ease access to personal (although public) data. One way to make such reverse engineering much harder to perform is to pack sparse encodings into low-dimensional embeddings: such dense representations are typically trained to reflect similarities between objects, so the original information is greatly corrupted and almost impossible to restore.

Since many of our sparse features have such dense counterparts to improve the performance of the system, we have conducted additional tests to see how much of the performance would have been lost if we only use the "safe" dense dimensions. The $F_{1\,micro}$ performance decreased for tasks *Category*, *Location*, *Age*, *Revenue* and *Music Skill* on average by 5.6%, while for tasks *Religion*, *Org. Size* and *Political Party* there was no statistically significant difference. The complete results are available online.[3] Given that profile features and hashtags have not been represented with a complementary embedding, we believe that a complete privacy-friendly dense representation can be developed without loss of performances. Not only such representation can be safely released in future for research purposes, but it will also allow the usage of machine learning algorithms not performing well on highly sparse input.

## 5.7 Related Work

**Type prediction** Type prediction is a well-studied task in the Semantic Web community. The goal is to significantly increase the coverage typically by reusing the data that is already present in the knowledge base. Latest studies on this topic cast the type prediction task as the multi-label classification problem.

Melo et al. (2016) introduce SLCN (Scalable Local Classifier per Node) showing, to the best of our knowledge, the current state-of-the-art performance for type prediction. They compare SLCN to the previous statistical and heuristics-based approach called SDType (Paulheim and Bizer, 2013) demonstrating improvements in all cases. Their

system extracts features from the entities in the knowledge graph and uses off-the-shelf classifiers to generate predictions. Then a special procedure is introduced to produce the inferred types. Rico et al. (2018) iterate on this idea trying to solve the partial depth problem of multi-class classifiers. As in Melo et al. (2016) they extract features from entities and use off-the-shelf classifiers, such as Naive Bayes and a densely connected neural network to infer types. Comparison against the SDType (Paulheim and Bizer, 2013) approach shows same or better results. Kejriwal and Szekely (2017) use low-dimensional dense representations of entities (embeddings) by Ristoski and Paulheim (2016a) to acquire representations for types in the same vector space. The resulting shared vector space allowed them to perform the type recommendation task, where for each entity a ranked list of topically relevant entities is produced. They show that entity embeddings can be utilized efficiently to infer type information. This idea was also suggested as a possible future direction in Ristoski and Paulheim (2016a).

To summarize, multi-label classifiers based on vector entity representations extracted from the knowledge graph are exhibiting state-of-the-art results for the type prediction task.

Here we design additional features that can be used in conjunction with the ones employed in these works. Similarly, Aprosio et al. (2013) brings the semi-structured data from different language chapters of Wikipedia to improve the type prediction performance. We provide the comparison to their features in Section 5.6.3.

**Embeddings** In Section 2.5, we provide a comprehensive discussion about different approaches that were proposed over the last years to produce *graph embeddings*. Many of the approaches described here can be used to represent DBpedia entities. Specifically the Trans-E (Bordes et al., 2013) approach is typically employed for the *link prediction* task, which is similar to the one presented here. Trans-E and other approaches exhibit varying performance in different tasks suggesting that there is no single superior method for representing graph embeddings. In this chapter, we employ the same approach (Cochez et al., 2017b) as the one used in Chapter 3 to represent entities in DBpedia. This approach exhibits comparable performance to its predecessor — RDF2Vec (Ristoski and Paulheim, 2016a; Ristoski et al., 2017) and is specifically designed to represent RDF graphs, such as DBpedia. Therefore, we employ the method by Cochez et al. (2017b) as a baseline for our experiments in this chapter. Additionally, we have tested different weighting schemas provided by the authors and chosen the "PageRank Split" variant as the optimal one for our task.

**Social media analysis** Social media analysis and user profiling, in particular, are being extensively researched topics both from the computer science and the societal standpoint. Over the years all possible combinations of features extracted from social media have

been explored including social graph, textual, image and video content, semi-structured profile data and various metadata. Most notably, Li et al. (2014) used all available user information to predict latent attributes of a user using weak supervision. Zheleva and Getoor (2009) have demonstrated the importance of social graph-based features by inferring hidden attributes of completely private profiles. Defense mechanisms to protect users from such inference have also been investigated (Nechaev et al., 2017a, Chapter 6).

To facilitate the usage of social media data in the Semantic Web environments, we have created the SocialLink project (see Nechaev et al., 2017b,d, 2018b, Chapter 3 and 4), further interlinking the LOD cloud and the social media. SocialLink enables more than tenfold increase in the number of entities which we can use for training and population, which will make our approach even more performant (by providing more training samples) and useful. Coverage comparison is provided in Table 5.5.

## 5.8 Conclusions

In this chapter, we showcase the use of social media data to perform entity type prediction on knowledge graphs. In particular, we show that Twitter data is able to complement existing RDF-based entity representations from DBpedia when used as input in the supervised type prediction setting. Our approach employs rich domain-independent representations derived from written text, social graph, and user profile attributes on Twitter, efficiently utilizing embeddings to further boost the performance and increase coverage. We demonstrate significant performance improvements (up to 11.7% $F_{1\,macro}$) of such hybrid approach on four different type prediction tasks compared to the performance of the state-of-the-art RDF-based representation by Cochez et al. (2017b).

By trading off the recall of such approach, the injection of social media data may allow expanding current ontology population efforts for knowledge bases, such as DBpedia, with the population of entity types from social media data. Moreover, more efficient and specialized machine learning techniques, such as SLCN (Melo et al., 2016), can be integrated in our approach to boost prediction results even further. In addition, we showed that Twitter data can be used in conjunction with Wikipedia text, significantly improving the performance in most of the considered prediction tasks.

# Chapter 6

# Concealing Interests of Passive Users in Social Media

User profiling has existed in the social media since their inception and has supported most of their business model. Even if users do not actively share the information about themselves on the social media (so-called passive users), they can still be profiled based on their location and who they follow. In this chapter, we present a system that leverages the links provided by the SocialLink dataset (Chapter 4) to help social media users to conceal their digital footprint. Specifically, our approach helps a passive Twitter user to stay private by proposing a list of additional profiles to follow that would confuse the social media's inference pipeline and prevent it from inferring useful information about that passive user and his interests. We demonstrate that SocialLink allows novel techniques to be developed that can protect user's digital identity from profiling.

## 6.1 Introduction

Currently, an enormous amount of people use social media every day: just recently, in July 2017, Facebook has hit two billion monthly users. Every action of those people is being recorded, analyzed and possibly sold to third parties in one form or another. Additionally, this data is used to acquire a digital footprint of users: what they like or do not like, their level of education, gender, race and much more.

Knowing that, people have learned to be careful about what they post, like or share on social media. Some go even further — they just follow a number of profiles they like and never actually generate any content that could be gathered or analyzed. In the literature, such users are called "passive users". A number of recent studies (Besel et al., 2016; Faralli et al., 2015b; Piao and Breslin, 2017; Zheleva and Getoor, 2009) have demonstrated that, despite their best effort, passive users can still be profiled based on location and the profiles they follow, information that is typically publicly available. The pipelines,

that exploit the list of followees to infer user interests, are based on the idea initially introduced by Besel et al. (2016). There it was demonstrated that the followee list could be mapped to a distribution of interest categories for the user by linking followees to their corresponding DBpedia/Wikipedia entries — a task that can greatly benefit from a resource like SocialLink— and then exploit the categorical information therein contained to derive an interests distribution for the target passive user. By adopting the paradigm "we are what we read", social media can infer digital footprint of passive users almost as good as for active ones, with the result that protecting the privacy of passive users remains an open issue.

Users can, of course, choose to stop using social media altogether in an attempt to preserve their privacy. However, in the modern world, it is becoming increasingly hard for an individual to abandon the benefits such services provide just for the sake of privacy. This situation is not unique to social media: the same happens with recent machine learning-based consumer products such as voice assistants, translation services and even self-driving cars. By using those services, people agree to provide data that is required for the correct operation of the system (e.g., to train it), but can also be used to perform inference of user profile attributes, in many cases, without the user even being aware of such usage.[1]

On the other hand, the privacy of the user is not something that has to be given up in favor of new exciting technologies and services. Companies can choose to protect user's privacy without degrading the user experience by implementing various techniques, such as differential privacy (Dwork, 2008) or by shifting the computation on the user-controlled device (Luo et al., 2009). Despite the availability of such techniques, modern social media are reluctant to implement them: their business depends on their ability to learn as much as possible about the target user and exploit this information the best they can to show advertisement and sell auxiliary services.

In this chapter, we present a system that helps Twitter passive users to conceal their digital footprint by leveraging the alignments from Twitter profiles to DBpedia entities provided by SocialLink (Nechaev et al., 2017b,d, 2018b, Chapter 4) and the categorical information available about those entities in their knowledge base entries. First, we show how to exploit SocialLink to create a state-of-the-art user's interest inference pipeline mirroring the approaches in the literature that use the list of followees (Besel et al., 2016; Piao and Breslin, 2017). Specifically, we use the high-quality alignments of SocialLink to map Twitter profiles to DBpedia resources, in place of the simple heuristics typically used in state-of-the-art approaches. We then use a custom mapping procedure to obtain the

[1] See, for example, the recent controversy regarding the usage of user's location (http://fortune.com/2019/01/04/la-ibm-weather-channel-app).

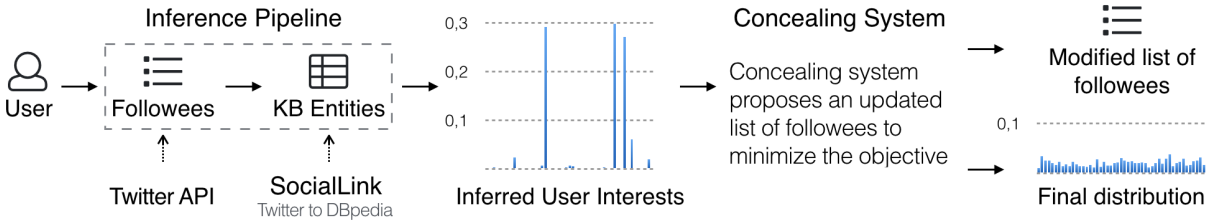



Figure 6.1. The proposed concealing approach

distribution across a taxonomy of 49 interest for each individual followee, which are finally combined to acquire the top interests for the target user.

Based on the interests inference pipeline build on SocialLink, our main contribution here are two concealing approaches (Greedy and Joint) that help passive users to stay private by proposing additional Twitter profiles to follow (followees) that would turn the user interests distribution inferred from followees close to the uniform one. This has the effect of confounding the social media's inference pipeline, preventing it from inferring useful information about the real interests of the target user. The original list of followees could then be stored on a user-controlled device (using a custom application or a browser plugin) allowing to recreate the original timeline. Our system proposes as few followees as possible to circumvent possible social media-induced limitations, reduce network load and cluttering of the user timeline. Our proposal is highlighted in Figure 6.1. This task of concealing user interests is inspired by the obfuscation examples provided in Brunton and Nissenbaum's book (Brunton and Nissenbaum, 2015), specifically, the "Bayesian Flooding" idea by Kevin Ludlow (Ludlow, 2012).

We evaluate our Greedy and Joint approaches and a Random baseline against our interest inference pipeline. We showcase a number of results, demonstrating that the Joint approach is able to achieve almost a perfect uniform distribution, decreasing the average KL-divergence by 94% compared to the Random baseline. The Joint approach solves a joint optimization problem learning the most efficient followee configuration. Additionally, we show the impact of our approaches on the performance of the inference pipeline using the precision at rank N (P@N) metric and, since we aim at suggesting as few new followees as possible, the average amount of suggested profiles.

Finally, we test our concealing approaches in a real world setting by evaluating them against the Twitter's *Who To Follow* (Gupta et al., 2013) recommendation system. This system recommends a target user other Twitter profiles to follow based (also) on his/her inferred interests, which can be deduced from produced recommendations leveraging the same DBpedia/Wikipedia alignments and categorical information used in the aforementioned inference pipeline. We show how our approaches can partially equalize

those inferred interests, proving that our techniques have general applicability and can be used with little or no knowledge about the target inference pipeline algorithm.

The rest of the chapter is structured as follows. In Section 6.2, we briefly present the related work. Then, we formally define our problem in Section 6.3, followed by the description of the user's interest inference pipeline used as reference in Section 6.4. Our concealing approaches are described in Section 6.5. Finally, we report the evaluation results in Section 6.6, and we conclude in Section 6.7.

## 6.2 Related Work

**User profiling** Profiling of users in social media has been performed since their inception both by the social media themselves and researchers. The inference of many user profile attributes, such as gender (Zheleva and Getoor, 2009), age (Rao et al., 2010), location (Sadilek et al., 2012), political affiliation (Zheleva and Getoor, 2009), level of education (Li et al., 2014), and occupational class (Preotiuc-Pietro et al., 2015), has been studied both for active (Abel et al., 2011; Michelson and Macskassy, 2010; Mislove et al., 2010; Piao and Breslin, 2016; Siehndel and Kawase, 2012; Zarrinkalam et al., 2016) and passive users (Besel et al., 2016; Faralli et al., 2015b; Piao and Breslin, 2017, 2018; Zheleva and Getoor, 2009).

Followee information (i.e., the social graph) of a user plays a key role in user profiling. One of the early studies successfully utilizing followee information to infer a user profile attribute was Sadilek et al. (2012). The usage of GPS data of followees allowed them to pinpoint the exact location (down to coordinates) of a target user with 80% accuracy. No additional information from the user's profile was used. However, their approach required a significant amount of high-quality location data to perform inference, which is in many cases hard to acquire.

Zheleva and Getoor (2009) exploited the social graph to infer gender and political affiliation of users. They were able to profile both active and passive users. Additionally, the more profiles of friends were public, the better the accuracy of such inference. Their study is one of the first that raised important questions about the user privacy in the social media. They concluded that the measures typically employed by social media to protect personal data are not sufficient.

More recently, there has been an increasing interest in profiling interests of passive users. Besel et al. (2016) proposed an inference pipeline that utilize followee information to infer interests. They link a target user's followees to the corresponding Wikipedia pages using their names and then Wikipedia category information is used to determine interests. Our inference pipeline is constructed following their approach. Piao and Breslin (2017)

iterated on this idea and produced a better performing system by improving the entity linking step and using a different interest propagation approach.

In summary, social graph-based features have proven to be useful in all cases, confirming the idea that "you are what you follow".

In most approaches for user profiling, a classifier is built for the inference of each individual attribute. However, more holistic approaches were studied as well. For example, Li et al. (2014) proposed a two-layered structure. On the first layer they reimplemented typical classifiers for attributes like location or education from previous works. Then they built a Probabilistic Logical Reasoning framework and used the results from the first layer as evidence. Since the accuracy of the first layer is not 100%, the second layer should be able to account for possible errors and handle contradictory knowledge, effectively preventing error propagation. Two reasoning approaches were explored: Markov Logic Networks and Probabilistic Soft Logic. As a result they were able to work with a wide variety of attributes: from gender to general like relationships towards entities.

**Privacy in the social media** Many user profiling studies cover the topic of users' privacy in social media by warning of the potential risks of sharing private information in user profiles. Privacy problems in social media, however, go beyond user profiling. Bettini and Riboni (2015), for instance, produced a comprehensive study on privacy preservation and the technological challenges arising in pervasive systems such as social media.

Felt and Evans (2008) studied how social media themselves can protect users by redesigning their APIs. They devised a privacy-by-proxy technique, where data is revealed to applications only when needed, limiting the scope to prevent data harvesting.

Luo et al. (2009) proposed to protect social media users by encrypting the user-related information before it reaches the social media. Their approach aims at achieving a goal similar to ours: not only to prevent third parties from accessing the sensitive data of the target user but also the social media themselves.

Ludlow (2012) introduced the concept of Bayesian Flooding demonstrating that the social media's advertisement and recommendation systems can be confused by flooding the user's timeline with artificial actions.

**Binarized Neural Networks** Even though we do not employ neural networks in this chapter, our Joint approach was influenced by recent studies on Binarized Neural Networks (BNN) (Bengio, 2013; Hubara et al., 2016). We used the binarization and back-propagation procedures from Hubara et al. (2016) to find an optimal solution to our optimization problem.

## 6.3 Problem Definition

The goal of this chapter is to protect the privacy of passive users by modifying their lists of followees in such a way that makes it much harder for the target inference pipeline to profile their interests. Followees are now being universally used by social media as part of the digital footprint of a person and play a crucial role in inferring user profile information such as interests. Even if the user does not post or share content on the social media (passive user), followee data is still available to third parties and the social network itself. The idea is to conceal this information without degrading the user experience, which, in case of modifying his/her followee list, can be achieved by storing the original unmodified list on a user-controlled device and use it to filter the timeline.

In this work we focus on concealing approaches tackling the inference of passive users' interests. Given a user $u$ and his/her list of followees $l_u$, a user's interests inference pipeline $g(l_u)$ is designed to infer this user's interests $\mathbf{c}^u = g(l_u), \mathbf{c}^u \in \mathbb{R}^n, c_i^u \geq 0, \sum_i c_i^u = 1$, where $n$ is the number of interest categories and $c_i^u$ is the score of interest category $i$ for user $u$. The categories are then ranked based on their score $c_i^u$ and the final list of top $k$ categories is produced to represent the user's interests. Real-world implementations of such inference pipeline will have a set of thresholds to abstain from classifying a user's interests when their ranking is too ambiguous, i.e., $c_i^u$ scores of top categories are very close to each other, making it impossible to reliably determine a user's interests.

An ideal approach for concealing user interests would try to make all the interest categories indistinguishable from each other. Therefore, the goal of our system is to modify the information about the user, i.e., transform user profile $u$ to user profile $u'$ (in our case produce a modified list of followees $l_{u'}$), so that the target inference system produces ambiguous results. Formally, the objective is to minimize the *Kullback-Leibler divergence*[2] between the $\mathbf{c}^{u'} = g(l_{u'})$ and the uniform distribution over possible interest categories:

$$D(u') = D_{KL}(\mathcal{U}\{1,n\}||\mathbf{c}^{u'}) = -\sum_i \frac{1}{n}\log c_i^{u'} + \sum_i \frac{1}{n}\log\frac{1}{n} = -\sum_i \frac{1}{n}\log c_i^{u'} - \log n \quad (6.1)$$

A possible limitation of such problem formulation is that the social network can impose limitations on the amount of follow requests from the user and a large list of followees can significantly increase the amount of API requests needed to acquire the timeline thus creating a worse user experience. In this case we may require our system to propose as few modifications to the initial followees list as possible and the final objective will be as follows:

$$J(u', u) = (1-\alpha)D(u') + \alpha(|l_{u'}| - |l_u|), \quad \alpha \in [0, 1) \quad (6.2)$$

[2] http://en.wikipedia.org/wiki/Kullback-Leibler_divergence

where $\alpha$ is a parameter that balances the tradeoff between minimizing the KL-divergence (which requires adding followees to equalize inferred interest categories) and minimizing the amount of proposed followees.

## 6.4 Interests Inference Pipeline

To develop our concealing approaches (presented in Section 6.5) we have implemented a user's interests inference pipeline ($g(l_u)$ in Section 6.3) that infers user's interests based on the list of followees. We follow Besel et al. (2016) state-of-the-art approach, improving it by removing dependencies from the Wikipedia API and the WiBi Taxonomy. We do that by replacing the Entity Linking heuristics used there with our state-of-the-art resource, SocialLink, which we have introduced in details in Chapter 4, and pre-computing the category distributions over a taxonomy of 49 top categories for all possible entities in English DBpedia/Wikipedia. This enables us to acquire a simple and robust system that allows testing different approaches for concealing a user's digital footprint. More in detail, the pipeline employs the following three-step procedure:

**Fetch followees** Followees $l_u$ of the target user $u$ are fetched using the Twitter API.

**Link followees to DBpedia/Wikipedia** Each followee profile $f \in l_u$ is linked using the SocialLink resource described in Chapter 4. Each alignment in this resource is associated with a confidence score $s_f$ that we use here to appropriately weight the contribution of each followee $f$ to the final user's interest distribution. We want to make sure that our linking procedure is robust since an error at this step will propagate along the pipeline. Therefore, we selected a subset of $m = 101,769$ high-quality alignments from SocialLink $v2.0$[3] by setting custom conservative thresholds on confidence scores (with respect to the default ones used in Chapter 4) that provide 91% precision and 31% recall performance against SocialLink gold standard.

**Produce interest scores** At this step each aligned followee has to be mapped to a category distribution $\mathbf{f}$, whose elements $f_i$ are the scores for interest categories $i$. Similarly to Besel et al. (2016), we use the DBpedia/Wikipedia category graph to propagate the specific categories associated to the followee entity up to the 49 top-level categories here considered, for each of which a relevance score is computed. This process resulted in a list of 3,507,016 scored DBpedia/Wikipedia entries. The interest scores in the category distribution $\mathbf{f}$ are then normalized using a modified softmax function $\sigma(\mathbf{f})$ to produce a

[3] SocialLink $v3.0$ was not available at the time our experiments were conducted.

valid probability distribution across possible interests, where the normalized score $\sigma(\mathbf{f})_i$ for interest category $i$ is computed as follows:

$$\sigma(\mathbf{f})_i = \frac{z(f_i)}{\sum_{k=1}^{n} z(f_k)}, \quad z(x) = \begin{cases} e^x & \text{if } x \neq 0 \ , \\ 0 & \text{if } x = 0 \end{cases} \tag{6.3}$$

This normalization procedure preserves zero scores for categories that were not observed for the given followee $f$, thus reducing the noise across final user's interests. The interests distribution $\mathbf{c}^u$ for the user $u$ is finally computed as a weighted average of normalized scores for user's followees $f \in l_u$:

$$\mathbf{c}^u = \frac{\sum_{f \in l_u} s_f \sigma(\mathbf{f})}{\sum_{f \in l_u} s_f} \tag{6.4}$$

The code of our interests inference pipeline along with the precomputed list of scores for each aligned followee can be found in our GitHub repository.[4]

## 6.5 Concealing Approaches

In order to conceal a user's interests we propose three approaches that calculate an updated list of followees to minimize the objective defined by (6.2): Greedy approach, Joint approach, and the baseline Random approach. In all cases, the system expects the initial list of followees in input and chooses new followees from the same list of pre-aligned profiles (from SocialLink) that our interests inference pipeline uses.

**Random approach** The most trivial direction we can take is to randomly follow new people in hope that the category distribution will become closer to uniform. In this approach, the profiles to follow are randomly drawn from the above-mentioned list and the system will stop when 250 new unique followees are selected. Since the list has its own bias towards certain categories, it can be expected that the more followees we add this way the closer we get to the list's own category distribution. This is why the threshold on the amount of new followees has to be selected carefully in order to provide a positive improvement in terms of our objective. Figure 6.2 shows how the average KL-divergence changes based on the amount of followees proposed.

**Greedy approach** The greedy approach iteratively selects a new followee from the pre-aligned list by picking the one that will decrease the KL-divergence between the resulting category distribution and the uniform distribution the most. Therefore, it can be seen as

[4] http://github.com/Remper/re-coding-ws

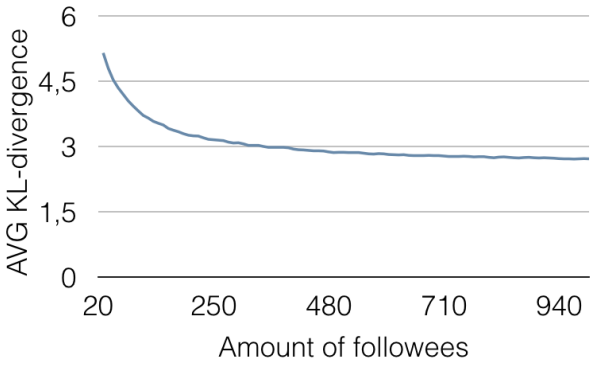



Figure 6.2. Average KL-divergence for different amounts of followees using Random approach

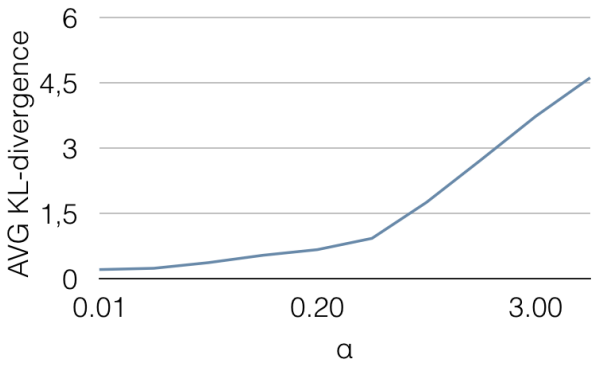



Figure 6.3. Average KL-divergence for different $\alpha$ values using Greedy approach

a breadth-first search over the space of possible configurations. The algorithm stops when it is no longer possible to select a new profile to follow that will improve the objective score (6.2). In our experiments the $\alpha$ parameter is set to 0.01. Figure 6.3 shows how the average KL-divergence changes based on the choice of $\alpha$.

**Joint approach** Finally, we have devised an approach that directly solves the formulated optimization problem by jointly finding an optimal followee configuration. This approach is inspired by recent studies about Binarized Neural Networks (Hubara et al., 2016) and it effectively "learns" the binary mask $\mathbf{w}^b$ where each element $w_j^b$ corresponds to a possible followee $j$ to add, being 1 if that followee $j$ should be followed and 0 otherwise.

Given the matrix of followees $F \in \mathbb{R}^{m \times n}$ that is obtained by simply stacking row-wise the category distributions $\sigma(\mathbf{f})$ of the pre-aligned followee list (i.e., $m = 101,769$) following the normalization procedure from (6.3), the $\mathbf{c}^u$ can be rewritten in terms of our binary mask $\mathbf{w}^b$:

$$\mathbf{c}^u = \frac{1}{\sum_i \mathbf{w}_i^b} \mathbf{w}^b F \tag{6.5}$$

The objective score can be computed as in (6.2). In order to solve this optimization problem, we define an additional weight vector $\mathbf{w} \in \mathbb{R}^m$, and $\mathbf{w}^b$ will now be computed using a simple deterministic binarization:

$$w_j^b = Bin(w_j) = \begin{cases} 1 & \text{if } w_j \geq 0 \ , \\ 0 & \text{if } w_j < 0 \end{cases} \tag{6.6}$$

Then $\mathbf{w}$ is learned by gradient descent towards the objective. Note that since the derivative of the *Bin* function is zero almost everywhere, the gradient have to be back-propagated using the *straight-through estimator technique* suggested by Bengio (2013). The $\mathbf{w}$ parameters are initialized by drawing from $\mathcal{N}(2l, l)$, where $l$ is the learning rate,

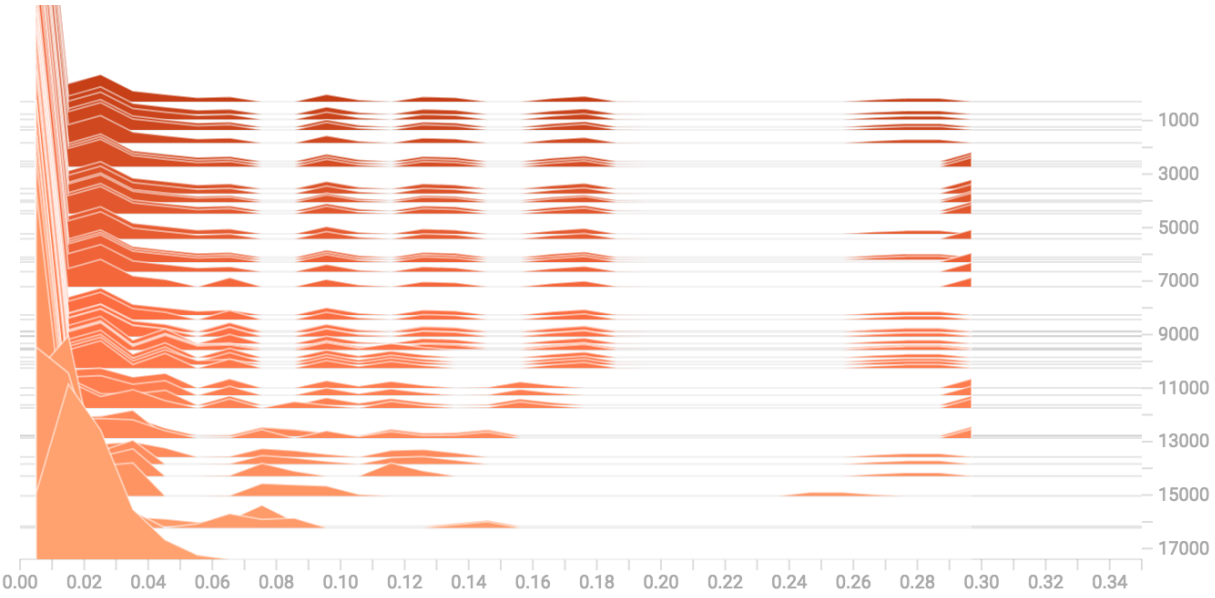



Figure 6.4. An example histogram of user's categories converging towards uniform distribution over 17k iterations of Joint approach. Each slice represent score distribution among categories for the corresponding iteration. In a perfect scenario all scores should be equal to $1/n = 0.02$.

ensuring that the majority of possible followees are followed at the first iteration. Figure 6.4 shows how a user's category distribution changes during the learning process.

## 6.6 Evaluation

We evaluate our concealing approaches Random, Greedy, and Joint against the user's interests inference pipeline of Section 6.4 and the Twitter's Who To Follow recommendation system. We measure four main performance metrics: (i) the average amount of followee suggestions; (ii) the ability to conceal a user's top-N interests categories, measured taking the categories produced without concealing as gold standard and measuring the precision P@N of the inference pipeline in reproducing those categories after concealing is applied; (iii) the average delta between the first and the second category probabilities; and (iv) the average KL-divergence between the category distribution produced by an approach and the uniform distribution across interest categories.

### 6.6.1 Evaluation against Interest Inference Pipeline

In order to produce the dataset required to evaluate our concealing approaches, we gathered a list of all the authors from ISWC proceedings from a three year period (2014-2016), extracting the names, titles and abstracts of all the papers associated with each author. After gathering the list, the Social Media Toolkit (see Chapter 7) was used to find the corresponding Twitter profile for each author, leveraging the collected textual context. This

Table 6.1. System evaluation against our interests inference pipeline

| Approach | Suggestions | System performance (P@N) Top 5 | Top 10 | Top 15 | Score diff to 2nd best | Diff from uniform (KL-div) |
|---|---|---|---|---|---|---|
| No mod | 0 | 1.0 | 1.0 | 1.0 | 0.25 | 7.52 |
| Random | 250 | 0.39 | 0.58 | 0.65 | 0.12 | 3.15 |
| Greedy | **27.49** | 0.52 | 0.55 | 0.56 | 0.03 | 0.20 |
| Joint | 76.61 | **0.37** | **0.45** | **0.49** | **0.02** | **0.16** |

resulted in a dataset of 491 Twitter accounts that were used to evaluate our approaches against the user's interests inference pipeline. The choice of the ISWC target audience was made to showcase how easy it is to profile people in a particular community just by having a set of names, keywords, or other related textual context.

The followees list for each author was gathered via the Twitter REST API and provided to our interest inference pipeline to produce an initial interest category distribution. Then, each approach was able to propose a modified list of followees and the category distribution was recalculated. Table 6.1 shows the resulting performance for each approach and the baseline numbers without applying any concealing technique. The baseline is assumed to be perfect at predicting user interests (have 100% precision at 5, 10 and 15 top categories) since our goal is to hide true user interests from the inference pipeline.

It can clearly be seen that, even though the Random approach already significantly reduces the KL-divergence, it takes a significant amount of suggestions to achieve this result. Random hides the initial top K categories of the user well, but produces a new top category (usually, Sport) that stands out and makes the target inference pipeline consistently producing false positives towards this category, which was not our goal.

The Greedy approach, on the other hand, produces an almost uniform distribution, while providing a relatively small amount of suggestions. It does not hide the top K categories as well as the Random approach, but the target system would have mostly likely abstained to infer user's interests given such category distribution. The results of this approach clearly show that if the concealing system is able to predict the expected score with high accuracy for an arbitrary followees list (in our case the approach had perfect information), it is possible to confuse the target inference pipeline.

Finally, the Joint approach is able to find a more efficient followee configuration than Greedy, producing the best results in almost all metrics at the cost of an increased suggestions count. The amount of suggestions can further be tuned by setting different values for the $\alpha$ parameter. However, we consider the current configuration to be a reasonable tradeoff.

| Approach | System performance (P@N) Top 5 | Top 10 | Top 15 | Score diff to 2nd best | Diff from uniform (KL-div) |
|---|---|---|---|---|---|
| No mod | 1.0 | 1.0 | 1.0 | 0.84 | 12.23 |
| Joint | 0.20 | 0.37 | 0.47 | 0.44 | 8.56 |

Table 6.2. System evaluation against Twitter's Who To Follow

### 6.6.2 Evaluation against Twitter's Who To Follow

In this scenario we evaluate the Joint approach, resulting the best in the previous evaluation, against the Who To Follow system used in production by Twitter to suggest users to follow based on a target user's profile (Gupta et al., 2013).[5] This system was chosen because it provides a simple way to collect user's interests as measured by Twitter. In order to evaluate against this system, we have created a number of new Twitter users providing as little initial information about the user as possible to the social network,[6] apart the lists of followees, so to simulate the behaviour of passive users.

After the creation of each account, the initial Twitter recommendations of users to follow were gathered. These users represent popular Twitter accounts that are always recommended to new Twitter users in a certain location. Then, a number of users were followed from the pre-aligned list with the intention to give a clear bias towards some interest category. After that, we gathered again the users to follow recommended by the network. Finally, the Joint approach was used to propose the modified list and the network's suggestions were gathered one more time. Overall, three lists of Twitter user recommendations were gathered, with the initial list acting as a filter to clean the other two lists from the location-based and general popularity-based suggestions.

The remaining two lists were mapped to distributions of interest categories using the formula presented in (6.4). Then the same metrics used in Section 6.6.1 were computed to evaluate results. Unfortunately, since the Who To Follow box had to be gathered manually and fresh Twitter accounts had to be created every time, the evaluation was significantly downscaled compared to what we initially have hoped to measure. However, it can clearly be seen in Table 6.2 that, even though the concealing approach does not have any information on how the target user's interests inference system works, it is often able to conceal the user's true category distribution.

In order to achieve better results, a training set can be gathered to modify the followee matrix $F$ using the same manual gathering approach employed during the evaluation. However, this is currently beyond the scope of this chapter.

[5] http://support.twitter.com/articles/227220

[6] All user accounts were created using fresh email accounts using an IP address that can be tracked down to Microsoft Azure cloud datacenter in Cheyenne, USA.

## 6.7 Conclusions and Future Work

In this chapter, we have presented an application of SocialLink related to the inference and concealment of the passive user's digital footprint on social media. Specifically, we have shown how the high-quality Twitter-DBpedia alignments provided by SocialLink can be used to design a state-of-the-art user interest inference pipelines, and based on the same alignments, we have proposed techniques for concealing the user's interests. We have shown that by using simple techniques together with the SocialLink resource and without degrading user experience, passive Twitter users can prevent the network or a third party system from inferring their interests based on the knowledge of who they follow. As our approach relies only on social graph information, which is present in any social media, we believe it can be generalized and ported to other platforms like Facebook and Instagram.

Even though the discussion about the privacy of users online has been a hot topic lately, social media are reluctant to implement industry standard techniques such as differential privacy and on-device computation, wanting instead to preserve their ability to sell ads and promote their services. In this situation, we believe it is increasingly important to explore various ways users can protect their digital identity.

# Chapter 7

# Social Media Toolkit

In this chapter, we describe the Social Media Toolkit (SMT) – a set of tools built on top of the SocialLink approach to facilitate testing and development of SocialLink as well as to enable its application in various use cases. Firstly, SMT implements solutions for two distinct Named Entity Linking (NEL) scenarios: the direct disambiguation of user mentions in tweets against DBpedia and the disambiguation of named entities on arbitrary text against Twitter. Both scenarios exploit either the SocialLink pipeline, described in Chapter 3, or the SocialLink resource as seen in Chapter 4. Secondly, SMT provides a convenient API and user interface to debug and test SocialLink and is able to work both with the Twitter REST API[1] data directly and with the indices built during the *data acquisition* phase of the SocialLink pipeline as the source of data.

In addition to describing the SMT capabilities, here we will showcase the two systems built relying on the SMT to implement their core functionality: the NEL pipeline that participated in EVALITA 2016 competition called MicroNeel and the social media management platform called Pokedem. Both are vivid examples of the solutions that could be built on top of the SocialLink pipeline in general and the SMT in particular.

The SMT is described in Section 7.1. In Section 7.2, we detail the MicroNeel pipeline and Section 7.3 showcases the Pokedem system. Finally, we offer discussion and future directions for the SMT toolset in Section 7.4.

## 7.1 System Description

SMT is a set of tools designed to support both the development and the production use of SocialLink and the related approaches. The list of SMT capabilities ranges from exposing the prebuilt SocialLink resource for downstream tasks to providing custom-built SocialLink-based pipelines designed to enable additional use cases.

[1] https://developer.twitter.com/en/docs/api-reference-index

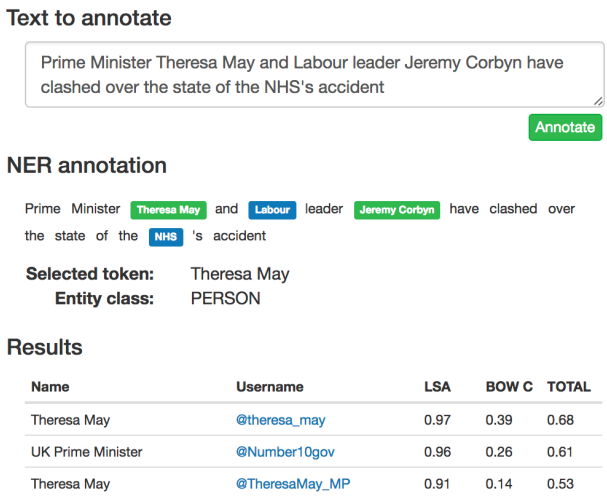



Figure 7.1. Debug UI of SMT built for testing the NEL functionality. This UI is also used to validate the SocialLink approach using NEL as a downstream task. The user inputs arbitrary text; the system highlights named entities using one of the NER backends; then given the selected token the API returns pairwise scores for each of the configured candidate selection models.

### 7.1.1 System API

The core module of SMT is the REST API written in Java that implements the main use cases of SocialLink pipeline and resource. SMT acts in two main capacities: it is used as a microservice to deliver the SocialLink capabilities to the downstream systems (like MicroNeel and Pokedem presented in Section 7.2 and 7.3 respectively); and it is used to manually debug and validate different setups of SocialLink, for example, using the UI shown in Figure 7.1 to facilitate such testing.

**Alignments** This part of the API consists of two methods that enable access to the deployed version of SocialLink resource. Both methods report pairwise entity-to-profile scores obtained during the *candidate selection* phase, providing alignments for an entity or a profile based on preselected thresholds.[2]

Firstly, the alignments/by_resource_uri method implements the regular forward query providing a list of SocialLink candidates for a given entity URI. This method is able to use owl:sameAs link to correctly resolve the entity URI and the candidates are returned along with their scores.

The second method, alignments/by_twitter_id, implements the reverse query to SocialLink resource: given a Twitter profile it lists all the entities that have this Twitter profile as a candidate. If the profile is linked to some entity, this entity will be appropriately marked. In case no entity is linked to a target profile, this method will also return the

[2]Thresholds include the *minimum score* for all versions of SocialLink and the *minimum improvement* for $v1.0$ and $v2.0$. Additionally, for $v3.0$ candidates are rescaled according to the procedure detailed in Section 3.4.1

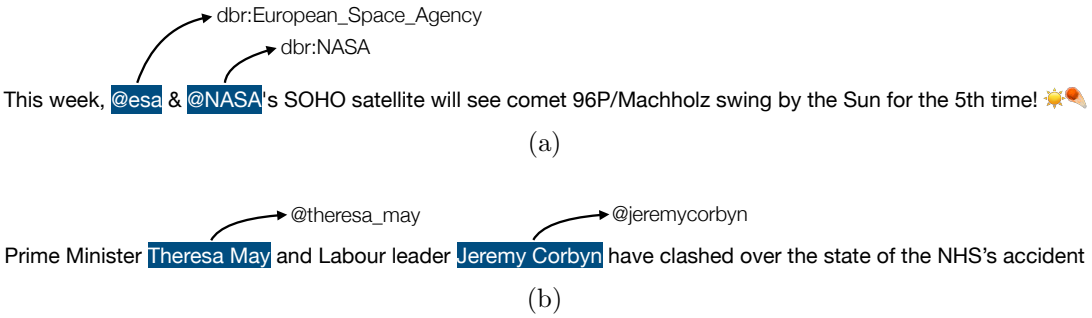



Figure 7.2. SMT API implements two NEL scenarios: direct disambiguation of mentions in tweets via the SocialLink resource and NEL on arbitrary texts against Twitter.

most probable type for this profile (either `PERSON`, `ORGANISATION` or `OTHER`) based on the list of candidate matching entities and their scores. This feature effectively allows users to disambiguate profile mentions in tweets against DBpedia directly or, at the very least, to acquire the most probable type for a mentioned profile. This method is employed to improve the NEL system called MicroNeel described in Section 7.2. Figure 7.2a showcases this capability.

**Annotation** Annotation toolset provides a set of NEL pipelines that rely on the SocialLink approach to link named entities in an arbitrary text to Twitter profiles. For example, as seen in Figure 7.2b, SMT is able to identify Theresa May and Jeremy Corbyn in a small passage as named entities and link them to their official Twitter accounts: @theresa_may and @jeremycorbyn respectively. This disambiguation from a surface form to a corresponding Twitter profile is implemented using a modified, generalized and customizable version of the SocialLink pipeline. This pipeline consists of the same three phases described in Chapter 3 – namely, *data acquisition*, *candidate acquisition*, and *candidate selection* – but builds on generalized notions of "user" and "entity". In SocialLink the user is the Twitter profile while the entity is the KB entry. In this NEL scenario addressed by SMT, the user is still the Twitter profile, but the entity is an object derived from the surface form in the text and its surrounding textual context (plus perhaps any available metadata that can be used as context). Endpoints described here are able to run multiple pipeline configurations enabling users to ensemble them into a better performing and robust system. For example, during the *candidate acquisition* phase it is possible to search candidates both using Twitter API and the User Index, expanding the list of candidates to increase the recall of the system when applied to less popular named entities.

The first annotation method of the API, annotate/ner, performs just the Named Entity Recognition (NER) using the configured backend. The second method, annotate/twitter, runs the complete three-phase pipeline using the preselected named entity as target for disambiguation and the rest of the text as context returning the sorted list of candidate

profiles alignments along with their scores. In addition to the provided SocialLink models, this method also runs a set of simple text-based approaches that act as baselines for testing purposes. Optionally, this method can return all the debug information reported by various subsystems: which textual tokens were matched against the vocabulary, similarity scores produced by similarity-based features, information about missing data, the order of candidate as returned by *candidate acquisition* phase, the number of filtered out candidates.

The annotate/twitter/simple method runs a simplified SocialLink pipeline (User Index-based candidate acquisition, single simplified candidate selection model) allowing greater disambiguation speed at the cost of less annotation results. This method is designed to work in production environments.

Finally, the "/" endpoint combines annotate/ner and annotate/twitter/simple disambiguating all entities of types `PERSON` and `ORGANISATION` in a given text.

**Misc** Additionally, SMT's REST API provides some quality-of-life tools for accessing Twitter including searching with topical filtering (profile/* methods), Twitter API mirroring (twitter/*) and additional debug methods, such as annotate/is_similar that allows comparing an entity and a Twitter profile in terms of textual similarity.

### 7.1.2 Configuration

SMT can be configured based on the requirements of the downstream tasks. The first configurable part is the choice of the tokenization and Named Entity Recognition system that given an arbitrary text would split it into tokens and, optionally, highlight named entities and their types. SMT supports two such systems out of the box: the Stanford CoreNLP (Manning et al., 2014) and the Wiki Machine.[3] The CoreNLP system, used by default, comes as a prepackaged dependency and doesn't require any additional configuration. While the Wiki Machine has to be installed and configured separately and the SMT would query its API to annotate the target text.

Secondly, as mentioned above, annotation toolset of SMT reuses the three-phase logic of the SocialLink pipeline, where each phase can either be a custom approach or taken as detailed in Chapters 3 and 4. Each phase can be independently configured as needed depending on a task and environment. SMT started as a test bench for SocialLink to perform live alignments using different combinations of features, input resources and neural models. The most basic setup requires neither the *Entity Index* nor the *User Index* to be present to perform the alignment. Additionally, the alignments API supports the usage of precomputed SocialLink alignments instead of live candidate acquisition and candidate selection steps.

[3] http://thewikimachine.fbk.eu/

**Data Acquisition** An entity in SMT is represented as a URI bundled with some amount of attributes describing it (context), while the user is a collection of Twitter API-compatible objects that typically include textual content (either written by the user or mentioning the user) and the user profile. This way the entity representation can accommodate a variety of data sources including RDF/OWL-based KBs as well as entities constructed from the arbitrary semi-structured data, like a named entity surface form with its context in a text. On the other hand, even though the SMT is designed to work with Twitter data, any other social media that offer similar functionality (named semi-structured profiles, a social graph, textual content) can be exploited. In other words, within SMT Twitter provides a reference format for user data that is general enough to accommodate user data from other social media.

Different SMT setups expect varying number of feature families to be produced from those entity and user representations. In its most complete form, SMT expects the following sources of features:

1. Entity attributes (either artificially constructed or queried from the specified knowledge base via its SPARQL endpoint based on entity URI):
   (a) Type information – using the rdfs:type property set to either dbo:Person, dbo:Organisation, dbo:Company or any other value to be considered of type `OTHER`.
   (b) Names – using foaf:name, foaf:givenName, foaf:surname and rdfs:label properties.
   (c) Entity descriptions – using dbo:abstract and rdfs:comment properties.
2. User attributes:
   (a) User profile – object compatible with Twitter user object.[4] Can be artificially constructed, acquired from the *User Index* or queried directly from Twitter REST API.
   (b) Real user context – a collection of tweets written by a user compatible with data format of the tweet object in Twitter.[5]
   (c) Estimated user context – preprocessed text written by a user derived from a stream of tweets and stored in the *User Index*.
   (d) Estimated social graph – social graph of the user approximated from a stream of tweets and stored in the *User Index*.

[4] https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/user-object
[5] https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/tweet-object

Besides the user profile, none of these sources of features are mandatory. The type information may be missing, in which case the entity will be considered of type `OTHER`. In case names for the entity are unavailable, the default name would be constructed from the entity URI (e.g., `http://dbpedia.org/resource/Barack_Obama` would have `Barack Obama` as name). The entity descriptions can be empty, meaning that no textual features would be available during the *candidate selection* phase. Real and estimated user contexts can be used together, either of which or even both can be empty. The estimated social graph is only required for the most complete model described in Chapter 3; without graph information, the approach would either default to average social graph embedding or ignore graph features based on configuration of the *candidate selection* phase.

**Candidate Acquisition** Two approaches are available in the SMT: live, based on Twitter search API queries, and index-based, relying on the *User Index*. The live approach implements the pipeline used in $v1.0$ of SocialLink: a single name-based search query is made to the search API using the top performing strategy, `Strict`, as described in our original paper (Nechaev et al., 2017b). The index-based approach is built upon the strategy detailed in Section 3.3.2 and used for versions $v2.0$ and $v3.0$ of SocialLink. Approaches can be used together, in which case the outputs are merged. Additionally, some variations of the SocialLink model use the explicit order provided by this phase. In case of the live approach, the order is returned by the Twitter API, while the index-based approach results are sorted based on a name-user pair frequency as observed from the stream of tweets. The order information is not available in case the two approaches are used together.

**Candidate Selection** Finally, SMT supports a multitude of different selection models based on available information gathered during the data acquisition phase. Candidate selection consists of two parts: feature extraction and prediction. Feature extraction produces a feature vector by either directly extracting features from the input data or populating features based on available user or entity identifiers using external resources. For example, given the textual information extracted from the initial entity and user representations, different dense or sparse vectorial representations can be produced based on the SMT configuration and to be used for the downstream prediction model. On the other hand, features, such as Wikipedia page statistics, social and knowledge graph embeddings are populated based on the respective identifiers. There are three basic feature extraction strategies that define three families of approaches implemented in the SMT:

- `ISWC17Strategy` implements Nechaev et al. (2017d) approach used in $v2.0$ of SocialLink. It defines basic feature families that exploit various similarities and categorical features derived from the available entity and user information corresponding to the BASE feature set as described in Table 3.3.

- `PAI18Strategy` implements the Nechaev et al. (2018b) approach used in $v3.0$ of SocialLink extracting the social graph and the knowledge graph-based features using the user and entity URIs.

- `SMTStrategy` extracts profile features described in Table 5.2 along with social graph embeddings and text embeddings derived from the provided textual contexts for the user and the entity. This strategy is specifically designed to require a bare minimum of information to be populated during data acquisition performing the least amount of feature engineering and streaming raw embeddings directly to the prediction model.

Each feature extraction strategy is configurable. For example, different precomputed word, social and knowledge graph embeddings can be used and some of the feature families (e.g., homepage matching or Wikipedia statistics) may be skipped.

Finally, the appropriate predictor model has to be trained. The training module of SMT is written in Python/Tensorflow. The model is typically trained on a DBpedia-based gold standard available on our website[6] using the same feature configuration as was employed during the feature extraction step. The training procedure involves conducting a full ten-fold cross-validation, out of which a production-ready ensemble model is produced averaging the scores provided by each of the trained models. Trained model is then exposed as a microservice allowing the rest of the SMT to query it supplying the unscaled feature vector to acquire the predicted score for each candidate-entity pair.

## 7.2 MicroNeel: A Tool to Perform Named Entity Detection and Linking on Microposts

In this section, we present the MicroNeel system for Named Entity Recognition and Entity Linking on Italian microposts that builds on the SMT system described in Section 7.1.1 and which was employed during our participation in the NEEL-IT task at EVALITA 2016. After performing a comprehensive preprocessing on input tweets, it merges SMT's annotations with the output of two state-of-the-art NLP tools, The Wiki Machine (Palmero Aprosio and Giuliano, 2016) and Tint (Palmero Aprosio and Moretti, 2016), using a custom rule-based or supervised approach. MicroNeel uses the SMT API to access the SocialLink resource alignments, employing it not only for the direct named entity disambiguation but also to determine the type of the target named entity using the populated candidates for each entity.

[6] https://w3id.org/sociallink#download

### 7.2.1 Background

Microposts, i.e., brief user-generated texts that include tweets, checkins, status messages, etc., are becoming an increasingly relevant source for information extraction. The application of Natural Language Processing (NLP) techniques to microposts presents unique challenges due to their informal nature, noisiness, lack of sufficient textual context (e.g., for disambiguation), and use of specific abbreviations and conventions like #hashtags, @user mentions, retweet markers and so on. As a consequence, standard NLP tools designed and trained on more 'traditional' formal domains, like news article, perform poorly when applied to microposts and are outperformed by NLP solutions specifically-developed for this kind of content (see, e.g., Bontcheva et al. 2013).

Recognizing these challenges and following similar initiatives for the English language, the NEEL-IT[7] task (Basile et al., 2016a) at EVALITA 2016[8] (Basile et al., 2016b) aims at promoting the research on NLP for the analysis of microposts in the Italian language. The task is a combination of Named Entity Recognition (NER), Entity Linking (EL), and Coreference Resolution for Twitter tweets, which are short microposts of maximum 140 characters (now doubled to 280) that may include hashtags, user mentions, and URLs linking to external Web resources. Participating systems have to recognize mentions of named entities, assign them a NER category (e.g., person), and disambiguate them against a fragment of DBpedia containing the entities common to the Italian and English DBpedia chapters; unlinked (i.e., NIL) mentions have finally to be clustered in coreference sets.

We participated to the NEEL-IT task with our MicroNeel system, which builds on SMT and allows us to showcase SMT's Named Entity Linking capabilities. In MicroNeel we investigated the use of two NER and EL tools on microposts – The Wiki Machine (Palmero Aprosio and Giuliano, 2016) and Tint (Palmero Aprosio and Moretti, 2016) – that were originally developed for more formal texts. To achieve adequate performances, we complemented them with: (i) a preprocessing step where tweets are enriched with semantically related text, and rewritten to make them less noisy; (ii) a NEL pipeline directly disambiguating user mentions in tweets, provided by the SMT; and (iii) rule-based and supervised mechanisms for merging the annotations produced by NER, EL, and SMT, resolving possible conflicts.

The Wiki Machine[9] is an open source Entity Linking tool that automatically annotates a text with respect to Wikipedia pages. The output is provided through two main steps: entity identification, and disambiguation. The Wiki Machine is trained using data extracted from Wikipedia and is enriched with Airpedia (Aprosio et al., 2013), a dataset built on top of DBpedia (Lehmann et al., 2015) that increases its coverage over Wikipedia pages.

[7] http://neel-it.github.io/
[8] http://www.evalita.it/2016
[9] http://thewikimachine.fbk.eu/

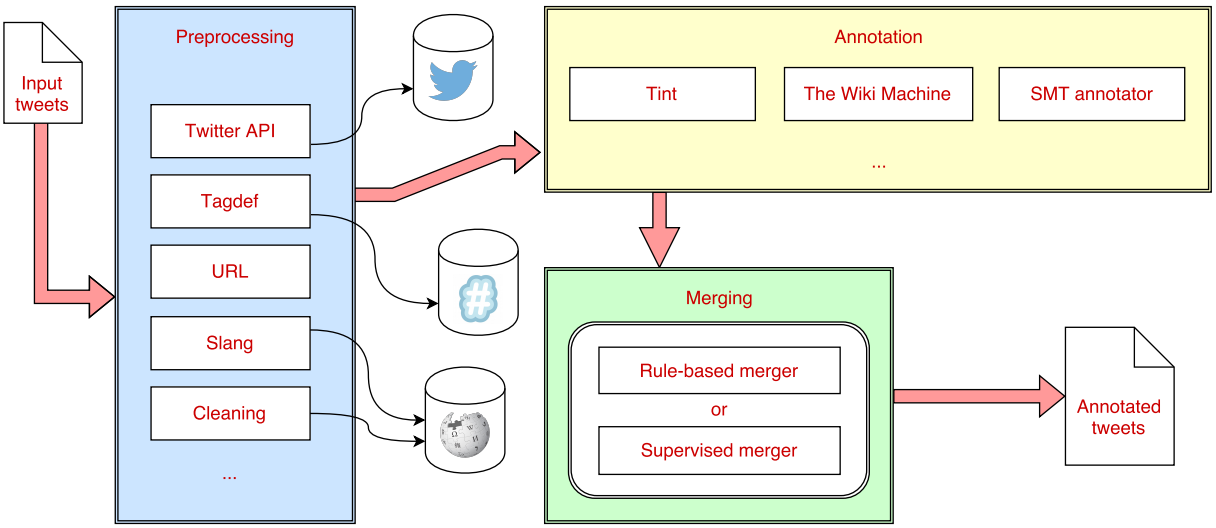



Figure 7.3. The overview of the system.

Tint[10] (Palmero Aprosio and Moretti, 2016) is an easy-to-use set of fast, accurate and extensible Natural Language Processing modules for Italian. It is based on Stanford CoreNLP[11] (Manning et al., 2014) and is distributed open source. Among other modules, the Tint pipeline includes tokenization, sentence splitting, part-of-speech tagging and NER.

For use in MicroNeel system, the custom instance of SMT was configured to expose the SocialLink resource $v1.0$ and it's alignments API is utilized to perform the reverse alignment directly disambiguating a user mention in tweet. As mentioned in Section 7.1.1, SMT is also able to classify any given Twitter profile as a person, organization, or other even if the alignment is not found in the resource. Additionally, SMT produces match probabilities allowing downstream merging approaches to take this confidence measure into account when blending different predictions together.

### 7.2.2 Description of the System

MicroNeel accepts a micropost text as input, which may include hashtags, mentions of Twitter users, and URLs. Alternatively, a tweet ID can be supplied in input (as done in NEEL-IT), and the system retrieves the corresponding text and metadata (e.g., author information, date and time, language) from Twitter API, if the tweet has not been deleted by the user or by Twitter itself.

Processing in MicroNeel is structured as a pipeline of three main steps, outlined in Figure 7.3: *preprocessing*, *annotation*, and *merging*. Their execution on an example tweet is shown in Figure 7.4.

[10] http://tint.fbk.eu/
[11] http://stanfordnlp.github.io/CoreNLP/

**Preprocessing** During the first step, the *original text* of the micropost is rewritten, keeping track of the mappings between original and rewritten offsets. The *rewritten text* is obtained by applying the following transformations:

- Hashtags in the text are replaced with their tokenizations. Given an hashtag, a bunch of 100 tweets using it is retrieved from Twitter. Then, when some camel-case versions of that hashtag are found, tokenization is done based on the sequence of uppercase letters used.
- User mentions are also replaced with their tokenizations (based on camel-case) or the corresponding display names, if available.
- Slangs, abbreviations, and some common typos in the text are replaced based on a custom dictionary (for Italian, we extracted it from the Wikipedia page `Gergo_di_Internet`[12]).
- URLs, emoticons, and other unprocessable sequences of characters in the text are discarded.
- True-casing is performed to recover the proper word case where this information is lost (e.g., all upper case or lower case text). This task employs a dictionary, which for Italian is derived from Morph-It! (Zanchetta and Baroni, 2005).

To help disambiguation, the rewritten text is then augmented with a textual *context* obtained by aggregating the following contents, if available:

- Hashtag descriptions from *tagdef*,[13] a collaborative online service;
- Twitter user descriptions for author and user mentions in the original text;
- Titles of web pages linked by URLs in the original text.

In the example shown in Figure 7.4, from the original tweet

> **[Original text]**
> (author: @OscardiMontigny)
> #LinkedIn: 200 milioni di iscritti, 4 milioni in Italia http://t.co/jK8MRiaS
> via @vincos

we collect

- metadata information for the author (Twitter user @OscardiMontigny);

[12] https://it.wikipedia.org/wiki/Gergo_di_Internet
[13] https://www.tagdef.com/

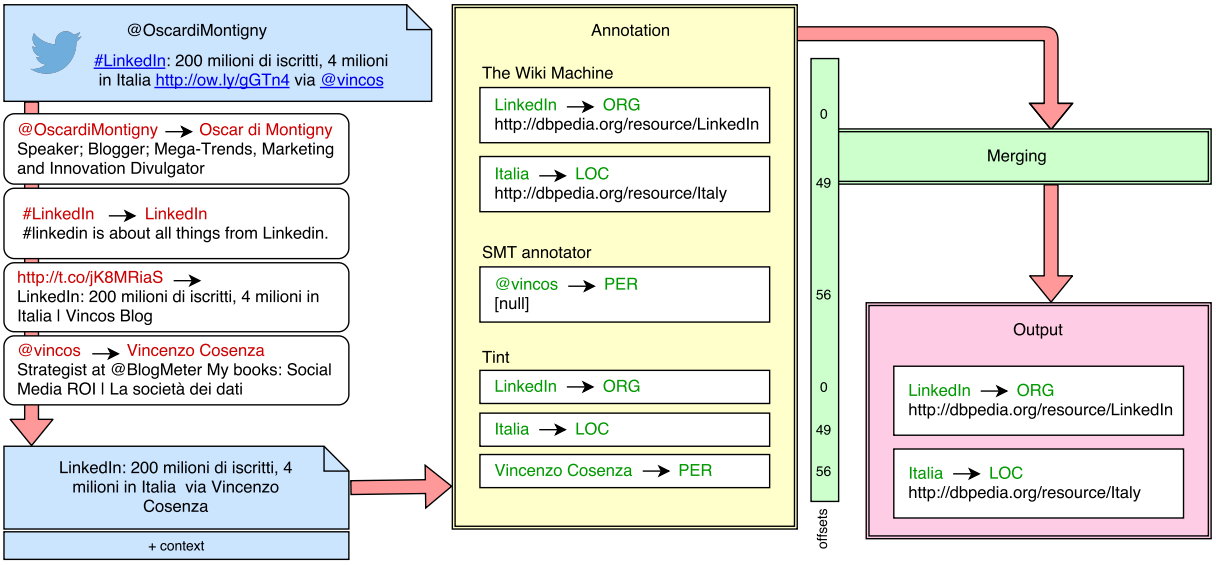



Figure 7.4. An example of annotation.

- description of the hashtag #LinkedIn;
- title of the URL http://t.co/jK8MRiaS;
- metadata information for the Twitter user @vincos, mentioned in the tweet.

The resulting (cleaned) tweet is

> [**Rewritten text**]
> LinkedIn: 200 milioni di iscritti, 4 milioni in Italia via Vincenzo Cosenza

with context

> [**Context**]
> Speaker; Blogger; Mega-Trends, Marketing and Innovation Divulgator. #linkedin is about all things from Linkedin. LinkedIn: 200 milioni di iscritti, 4 milioni in Italia — Vincos Blog. Strategist at @BlogMeter My books: Social Media ROI — La società dei dati.

**Annotation** In the second step, annotation is performed by three independent annotator tools run in parallel:

- The rewritten text is parsed with the NER module of Tint. This processing annotates named entities of type person, organization, and location.
- The rewritten text, concatenated with the context, is annotated by The Wiki Machine with a list of entities from the full Italian DBpedia. The obtained EL annotations are enriched with the DBpedia class (extended with Airpedia), and mapped to the considered NER categories (person, organization, location, product, event).

- The user mentions in the tweet are assigned a type and are linked to the corresponding DBpedia entities using SMT; as for the previous case, SMT types and DBpedia classes are mapped to NER categories. A problem here is that many user mentions classified as persons or organizations by SMT are non-annotable according to NEEL-IT guidelines.[14] Therefore, we implemented two strategies for deciding whether to annotate a user mention: the *rule-based SMT annotator* always annotates if the SMT type is person or organization, whereas the *supervised SMT annotator* decides using an SVM classifier trained on the development set of NEEL-IT.

The middle box in Figure 7.4 shows the entities extracted by each tool: The Wiki Machine recognizes "LinkedIn" as organization and "Italia" as location; SMT identifies "@vincos" as a person; and Tint classifies "LinkedIn" as organization and "Italia" and "Vincenzo Cosenza" as persons.

**Merging** The last part of the pipeline consists in deciding which annotations have to be kept and which ones should be discarded. In addition, the system has to choose how to deal with conflicts (for example inconsistency between the NER class produced by Tint and the one extracted by The Wiki Machine).

Specifically, the task consists in building a *merger* that chooses at most one NER class (and possibly a compatible DBpedia link) for each offset of the text for which at least one annotator recognized an entity. For instance, in the example of Figure 7.4, the merger should ignore the annotation of @vincos, as it is not considered a named entity.

As baseline, we first developed a *rule-based merger* that does not discard any annotation and solves conflicts by majority vote or, in the event of a tie, by giving different priorities to the annotations produced by each annotator.[15]

We then trained a *supervised merger* consisting of a multi-class SVM whose output is either one of the NER categories or a special NONE category, for which case we discard all the annotations for the offset. The classifier is trained on the development tweets provided by the task organizers, using libSVM (Chang and Lin, 2011) with a polynomial kernel and controlling precision/recall via the penalty parameter $C$ for the NONE class. Given an offset and the associated entity annotations we use the following features:

- whether the entity is linked to DBpedia;
- whether the tool $x$ annotated this entity;

[14] Basically, a user mention can be annotated in NEEL-IT if its NER category can be determined by just looking at the username and its surrounding textual context in the tweet. Usernames resembling a person or an organization name are thus annotated, while less informative usernames are not marked as their nature cannot be determined without looking at their Twitter profiles or at the tweets they made, which is done instead by SMT.

[15] Tint first, followed by The Wiki Machine and SMT.

Table 7.1. MicroNeel performances on NEEL-IT test set for different configurations.

| Configuration | Mention CEAF | | | Strong mention match | | | Strong link match | | | Overall |
|---|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 | F1 |
| *base* run | 0.514 | 0.547 | 0.530 | 0.457 | 0.487 | 0.472 | 0.567 | 0.412 | 0.477 | 0.497 |
| *merger* run | 0.576 | 0.455 | 0.509 | 0.523 | 0.415 | 0.463 | 0.664 | 0.332 | 0.442 | 0.475 |
| *all* run | 0.574 | 0.453 | 0.506 | 0.521 | 0.412 | 0.460 | 0.670 | 0.332 | 0.444 | 0.474 |
| *base* - NER | 0.587 | 0.341 | 0.431 | 0.524 | 0.305 | 0.386 | 0.531 | 0.420 | 0.469 | 0.429 |
| *base* - SMT | 0.504 | 0.525 | 0.514 | 0.448 | 0.468 | 0.458 | 0.564 | 0.372 | 0.448 | 0.477 |
| *base* - EL | 0.487 | 0.430 | 0.457 | 0.494 | 0.437 | 0.464 | 0.579 | 0.049 | 0.090 | 0.349 |
| *base* - rewriting | 0.554 | 0.399 | 0.464 | 0.492 | 0.356 | 0.413 | 0.606 | 0.354 | 0.447 | 0.444 |
| *base* - context | 0.513 | 0.547 | 0.530 | 0.453 | 0.485 | 0.468 | 0.566 | 0.416 | 0.480 | 0.496 |

- whether the tool $x$ annotated the entity with category $y$ ($x$ can be Tint, SMT, or The WikiMachine; $y$ can be one of the possible categories, such as person, location, and so on);
- the case of the annotated text (uppercase initials, all uppercase, all lowercase, etc.);
- whether the annotation is contained in a Twitter username and/or in a hashtag;
- whether the annotated text is an Italian common word and/or a known proper name; common words were taken from Morph-It!, while proper nouns were extracted from Wikipedia biographies;
- whether the annotated text contains more than one word;
- frequencies of NER categories in the training dataset of tweets.

The result of the merging step is a set of NER and EL annotations as required by the NEEL-IT task. EL annotations whose DBpedia entities are not part of the English DBpedia were discarded when participating in the task, as for NEEL-IT rules. They were however exploited for placing the involved entities in the same coreference set. The remaining (cross-micropost) coreference annotations for unlinked (NIL) entities were derived with a simple baseline that always put entities in different coreference sets.[16]

**Implementation** The MicroNeel extraction pipeline is available as open source (GPL) from the project website.[17] It is written in Java and additional components for preprocessing, annotation, and merging can be easily created by implementing an `Annotator` interface. The configuration, including the list of components to be used and their parameters, can be set through a specific JSON configuration file.

[16] It turned out after the evaluation that the alternative baseline that corefers entities with the same (normalized) surface form performed better on NEEL-IT test data.

[17] https://github.com/fbk/microneel

### 7.2.3 Results

Table 7.1 reports on the performances obtained by MicroNeel at the NEEL-IT task of EVALITA 2016, measured using three sets of Precision (P), Recall (R), and F1 metrics (Basile et al., 2016a):

- *mention CEAF* tests coreference resolution;
- *strong typed mention match* tests NER (i.e., spans and categories of annotated entities);
- *strong link match* assesses EL (i.e., spans and DBpedia URIs of annotated entities).

Starting from their F1 scores, an overall F1 score was computed as a weighted sum (0.4 for mention CEAF and 0.3 for each other metric).

MicroNeel was trained on the development set of 1000 annotated tweets distributed as part of the task, and tested on 300 tweets. We submitted three runs (upper part of Table 7.1) that differ on the techniques used – rule-based vs supervised – for the SMT annotator and the merger:

- *base* uses the rule-based variants of the SMT annotator and the merger;
- *merger* uses the rule-based SMT annotator and the supervised merger;
- *all* uses the supervised variants of the SMT annotator and the merger.

In addition to the official NEEL-IT scores, the lower part of Table 7.1 reports the result of an ablation test that starts from the *base* configuration and investigates the contributions of different components of MicroNeel: The Wiki Machine (EL), Tint (NER), SMT, the tweet rewriting, and the addition of textual context during preprocessing.

Contrary to our expectations, the *base* run using the simpler *rule-based SMT* and *rule-based merger* performed better than the other runs employing supervised techniques. Table 7.1 shows that the contribution of the *supervised SMT* annotator was null on the test set. The *supervised merger*, on the other hand, is only capable of changing the precision/recall balance (which was already good for the *base* run) by keeping only the best annotations. We tuned it for maximum F1 via cross-validation on the development set of NEEL-IT, but the outcome on the test set was a decrease of recall not compensated by a sufficient increase of precision, leading to an overall decrease of F1.

The ablation test in the lower part of Table 7.1 shows that the largest drop in performances results from removing The Wiki Machine, which is thus the annotator most contributing to overall performances, whereas SMT is the amounts to a +0.0193 F1. Even though the contribution of the SMT backed by the SocialLink is smaller compared to other annotators, it is to be expected given that the other two tools are specifically designed to perform NEL. The rewriting of tweet texts accounts for +0.0531 F1, whereas the addition of textual context had essentially no impact on the test set, contrarily to our expectations.

An error analysis on the produced annotations showed that many EL annotations were not produced due to wrong word capitalization (e.g., lower case words not recognized as named entities), although the true-casing performed as part of preprocessing mitigated the problem. An alternative and possibly more robust solution may be to retrain the EL tool not considering letter case.

### 7.2.4 Discussion

MicroNeel obtained the second best result in the NEEL-IT task at EVALITA 2016. In particular, MicroNeel got the best performance in the linking task, mainly thanks to the Wiki Machine and the additional EL contribution by SMT. Overall, these results demonstrate that MicroNeel approach is effective even if it builds on standard NER and EL components. The added value consists in the way these components are integrated and in the use of SMT to directly solve the NER + EL task for Twitter user mentions. In the future, we plan to revise the SMT annotator to exploit the latest version of SocialLink, to adapt MicroNeel to English and other languages, and to integrate some other modules both in the preprocessing and annotation steps, such as the NER system expressly developed for tweets described by Minard et al. (2016).

## 7.3 Pokedem: an Automatic Social Media Management Application

Another system that exploits SMT's API is Pokedem (Corcoglioniti et al., 2017, 2018) — an automatic social media management application for Twitter that we are developing and that leverages many of the results and expertise acquired within this thesis.

Typically, the task of managing the social media presence of a company or a public person is the job of a dedicated social media account manager. While many attempts have been made in recent years to provide more automation to account managers, complete workflow automation has still to be achieved.

Pokedem[18] is an application for Twitter that aims at filling this gap, recommending actions (tweet/retweet, follow/unfollow, like/unlike) that could be performed by the account manager with the business-relevant goal (Paniagua and Sapena, 2014) of improving account popularity and audience engagement, rather than suggesting people and contents that the user may like as typically considered in Twitter recommendation literature (Kywe et al., 2012). Beyond recommendations, Pokedem also provides account managers with comprehensive data analytics about the performance of the account as well as the profiling

[18] Demonstration video at http://pokedem.futuro.media/

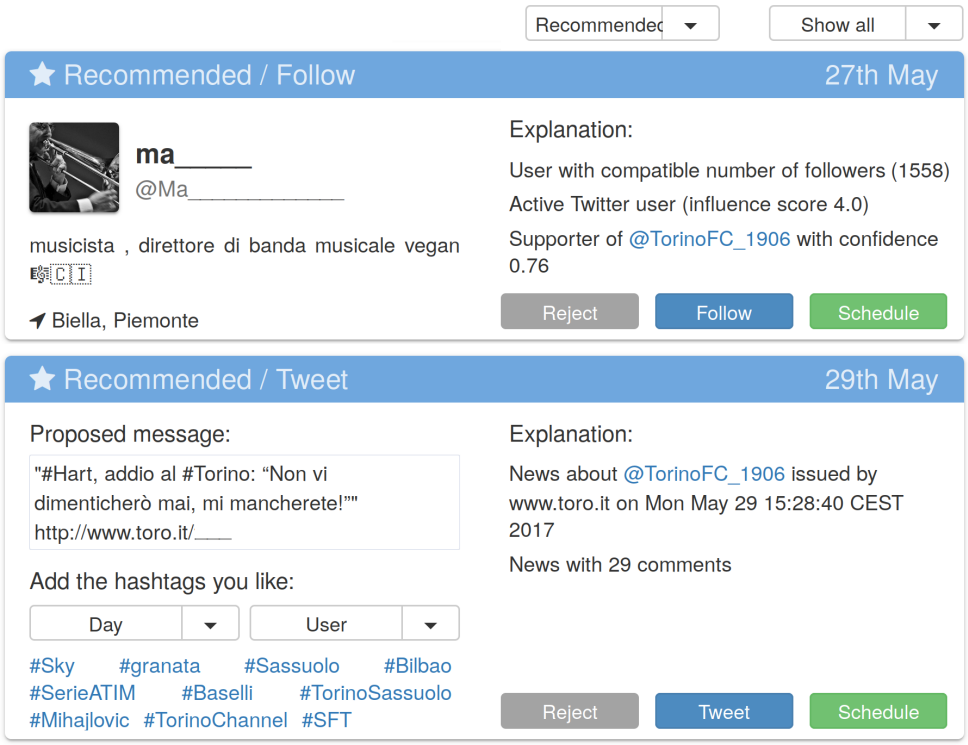
of the target audience, ultimately reducing the burden, required time and skills for account managers, and allowing them to focus on activities needing human judgment and creativity.

Pokedem relies on SMT to enrich the actions suggested, bolstering the target account performance. In particular, SMT is used to substitute named entities in the suggested text with the resolved Twitter profiles as proposed by the SMT pipeline. Additionally, parts of the feature extraction pipelines developed as part of the SocialLink are used to build user profiling approaches used by the system. In this section, we showcase the capabilities of Pokedem, highlighting through a use case deployment (@esseredeltoro account) how its use allows managing social media presence with little effort.

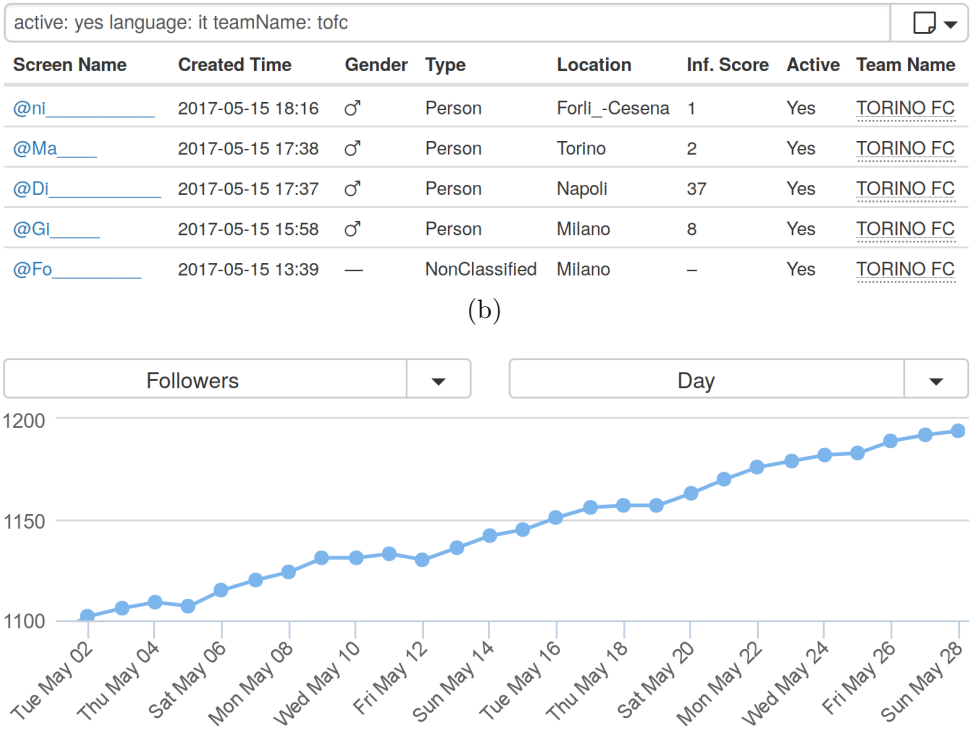
### 7.3.1 Background

Social media presence is recognized as an important factor for businesses, as it provides a channel for reaching potential customers, enabling for instance to gather their preferences and feedback, increase revenues through social marketing, and in general improve brand awareness and reputation (Paniagua and Sapena, 2014). However, maintaining a social media account and growing it by acquiring and engaging followers — the potential customers — are not for free. On the one hand, a successful account must provide some value to its followers, e.g., in terms of posted contents, conversation, and engagement. On the other hand, unless there is a reputation capital to leverage (e.g., a famous brand) and excluding the questionable practice of buying fake followers, gathering valuable followers that may interact and provide value to the account (differently from fake followers) may not be easy: these users have to discover the account first, so they must be actively searched, engaged and converted into followers by the account manager. All these activities require time and skills, and may be economically non-viable for professionals and small businesses lacking the required resources. While social media account automation tools exist to help account managers, they typically cover only the most basic tasks, like scheduling and optimizing posts for different social networks, and the feasibility and potential of further account automation are still largely unexplored from both a research and application points of view.

Many strategies are used to acquire followers on social media. Ignoring the paid promotion of one's account via the social media platform, and the purchase of fake followers (Cresci et al., 2015) that generally violates terms of use and brings no interaction (acquired followers are fake "zombie" accounts), these strategies generally sum up to (i) providing original contents and other value that may attract users, and (ii) actively engaging users, e.g., by following or mentioning them, to convert them into followers. For the latter strategies, the choice of the users to engage is crucial. Mass follow strategies typically (and controversially) pick random users, but even if they convert into followers




(a)




| Screen Name | Created Time | Gender | Type | Location | Inf. Score | Active | Team Name |
|---|---|---|---|---|---|---|---|
| @ni_____ | 2017-05-15 18:16 | ♂ | Person | Forli_-Cesena | 1 | Yes | TORINO FC |
| @Ma____ | 2017-05-15 17:38 | ♂ | Person | Torino | 2 | Yes | TORINO FC |
| @Di_____ | 2017-05-15 17:37 | ♂ | Person | Napoli | 37 | Yes | TORINO FC |
| @Gi_____ | 2017-05-15 15:58 | ♂ | Person | Milano | 8 | Yes | TORINO FC |
| @Fo_____ | 2017-05-15 13:39 | — | NonClassified | Milano | – | Yes | TORINO FC |

(b)




(c)

Figure 7.5. Pokedem web UI (exemplified for the @esseredeltoro Twitter account): (a) Recommendations tab, (b) User Profiling tab, (c) Analytics tab.

they may not be interested in interacting with the account, making them of limited value from a social and business points of view. On the other hand, manually picking users is time-consuming and does not scale. A *recommendation system* suggesting potential followers to engage – like the one provided by Pokedem also leveraging SMT– would thus be an invaluable asset for social media account managers.

The use of recommendation techniques to suggest the users to follow on social media has already been studied in the literature. However, the proposed approaches focus on recommending followees that a target/active user may find interesting. These approaches typically leverage either the social network topology via collaborative filtering techniques (Armentano et al., 2012; Kim and Shim, 2014; Zhao et al., 2013), the topics and features of user-generated contents via content-based techniques (Armentano et al., 2013), or both kinds of features (Hannon et al., 2010; Barbieri et al., 2014); some approaches also consider the user sentiment (Yuan et al., 2014) and personality traits (Tommasel et al., 2016). The task solved by these approaches differ from the recommendation task implemented by Pokedem as they do not aim at recommending users that may follow-back the target account, although in principle one may apply such systems *indirectly* by providing a ranked list of recommendations for *every* user in the social network (which may be millions) and then rank those users based on how high our target account looking for followers appears in those lists or how high it is scored (a ranking that we may assume to correlate with the follow-back likelihood). This is however an impractical solution if the goal is to provide recommendation to a single (or few) target account(s), and we are thus not investigating it further.

Other approaches in the literature take a network-centric view in recommending followees, e.g., by aiming at maximizing the content spread on the network (Chaoji et al., 2012), and thus address a task different from ours. Twitter itself provides a followee recommendation service (Gupta et al., 2013) that operates by default on each user homepage and is accountable for a large fraction of the follow relations on Twitter. A survey of recommendation tasks and systems on Twitter is provided by Kywe et al. (2012).

### 7.3.2 Description of the System

Pokedem consists of three main components, corresponding to distinct tabs in the web interface it provides to social media account managers (Figure 7.5): *Recommendations*, *User Profiling*, and *Analytics*.

**Recommendations** This tab (excerpt in Figure 7.5a) provides a ranked, always up-to-date list of recommended social media actions, with explanations of why they are recommended. The account manager may reject, execute immediately, or schedule a recommended action for later execution at an optimal time chosen by the system.

Recommendations are generated using *content-based recommendation techniques* (Lops et al., 2011) based on rich *user profile* features, tailored to each type of action, and aimed at improving account reputation rather than mimicking the account manager's behavior. Therefore, actions are proposed based on the likelihood of a positive feedback on social media—follow-back for follow or like actions, likes or retweets for tweet actions—estimated from previous actions done by the account. The execute/reject feedback of the account manager for past actions is currently ignored. Pokedem recommends:

- following users that may follow back the account (top box in Figure 7.5a), based on several user profile features that are compared in a content-based way with the ones of followers;
- tweeting links to relevant news articles (bottom box in Figure 7.5a), chosen from configurable RSS feeds based on news popularity (e.g., number of comments) and topicality;
- tweeting different charts comparing the social media performances (activity level, influence) of users on configurable topics, to engage them and improve followers retention.

For tweeting actions, we also recommend trending hashtags to increase tweet visibility. Additionally, Pokedem supports replacing named entities with corresponding Twitter accounts using the SMT NEL capabilities. This is done to both engage with the entities mentioned in the target tweet and to provide followers with rich content specifically tailored for Twitter.

**User Profiling** This tab (excerpt in Figure 7.5b) shows the rich user profiles collected by Pokedem to support recommendations. Profiles are computed for the target Twitter audience of the account, identified by navigating the social network from representative seed accounts. In addition to basic attributes (e.g., language, creation date, followers count), the following attributes are extracted from public social media data via state-of-the-art classification techniques and the linking of popular Twitter accounts to Wikipedia/DBpedia using the SocialLink pipeline and resource:

- user gender/type (person, org.), via rules and gazetteers;
- user location, either gathered explicitly from the user or estimated via supervised classifiers trained on explicit data;
- user influence score, consisting in a *h-index* like measure based on the number of user tweets and their retweets/likes;

- user activity level, based on number and recency of tweets;
- domain-specific attributes like the user's favourite football team, estimated mainly based on followed accounts.

Filtering facilities and attribute distribution charts support demographic analyses and *microtargeting* (Barbu, 2014), i.e., the selection and engagement of specific users on the basis of their attributes.

**Analytics** This tab (graph example in Figure 7.5c) leverages the metrics gathered for recommending actions (e.g., numbers of followers, likes, retweets) to offer charts and other analytics services. They allow comparing the performances of the managed account and of competitor accounts (listed by the account manager) along time and different dimensions, to identify the metrics needing improvement.

### 7.3.3 Results

We deployed Pokedem to @esseredeltoro[19], a Twitter account that we created on August 2016 to support the development and testing of Pokedem and that we use here to demonstrate its capabilities.

**Account** @esseredeltoro acts as a non-personal, community-like account targeting the supporters of Torino Football Club, a professional football team playing in the Italian top football division (Serie A). Similarly to other "competitor" Twitter accounts, @esseredeltoro aims at becoming a popular account among Torino's supporters, acquiring followers and fostering interaction with them.

**Challenge** As @esseredeltoro account managers needing followers and reputation starting from scratch, we look for interesting news to publish and people to follow. However, surfing social networks and websites to find good contents, and going over users accounts who might follow @esseredeltoro and spread its contents, are time consuming tasks. Deciding when to tweet to maximize impressions and interacting with users are other expensive tasks.

**Strategy** We delegated to Pokedem all the activities that could be automated: (i) keeping track of the latest Twitter trending topics related to Torino; (ii) recommending interesting and influential people to follow; (iii) discovering and recommending relevant contents about the team; (iv) creating and recommending charts about Torino's supporters most active on Twitter and their comparison with supporters of the opposite team in a match; and (v) assessing the impact of actions performed on the account. This let us concentrate

[19] https://twitter.com/esseredeltoro

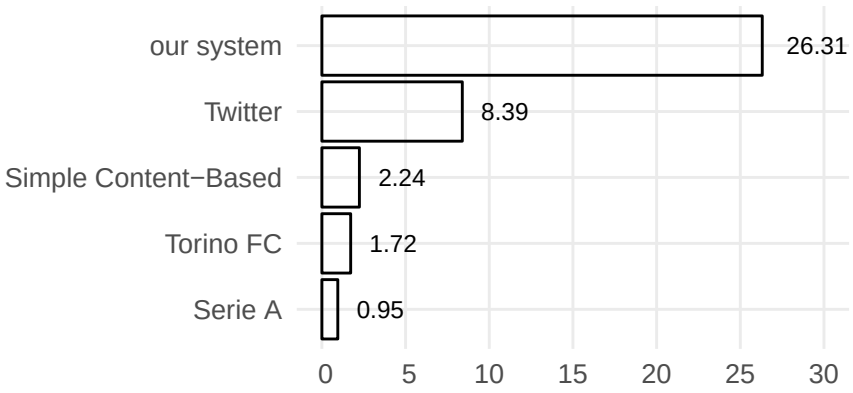



Figure 7.6. Conversion rates (percentage of recommended users converted into followers) of Pokedem compared to some of the accounts in the same domain (Serie A and Torino FC) and two simple baselines: a basic content-based follow strategy and the Twitter suggestions (Gupta et al., 2013)

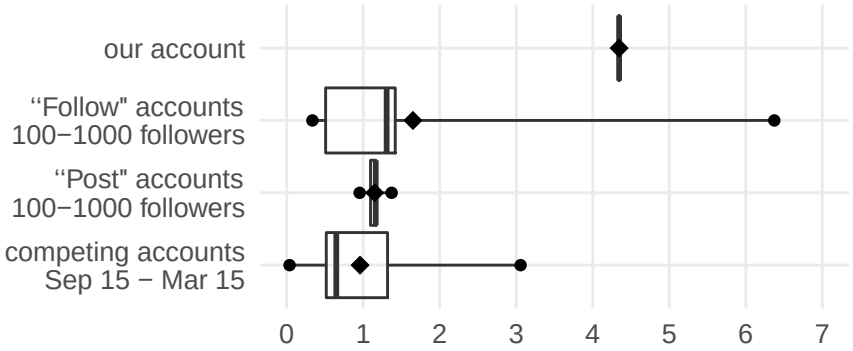



Figure 7.7. Growth rates (avg. new followers per day) on our account and the baselines for growing from 100 to 1000 followers and for period Sept. 15, 2016 – March 15, 2017.

on tasks requiring creativity and reasoning, like creating original contents and interacting with people that we follow or like.

**Results** ~50% of tweets and ~95% of follow actions done so far by @esseredeltoro were recommended by Pokedem, with considerable time savings for the account managers. 26% of users followed via Pokedem recommendations followed back (vs. 8% for Twitter recommendations (Gupta et al., 2013), tested by executing all recommended actions for 1 month). Conversion rates are shown in Figure 7.6. Together with the engaging posts recommended by Pokedem, this allowed @esseredeltoro to jump from 100 to 1000 followers in ~6 months (4.3 new followers/day, vs 1.0 avg. of competitors for the same 100 to 1000 followers growth), approaching competitors in terms of popularity and interaction rates (0.015 replies+retweets+likes/follower/day, vs 0.019 avg. of competitors). Figure 7.7 details the performance of Pokedem compared to various accounts exhibiting similar follow/post behaviour in our domain. "Follow" accounts tried to follow audience in the domain to acquire a follow-back, while "Post" accounts were focused on posting original

content to attract new followers. Finally, the figure also reports the growth rate of all competing accounts regardless of the chosen strategy.

### 7.3.4 Discussion

Pokedem can be deployed on different accounts to effectively improve reputation and audience engagement with its recommendations. Pokedem uses SocialLink feature extraction pipeline to build a user representation as well as SMT NEL capabilities to enrich the tweet before posting. A preliminary evaluation on a real Twitter account shows that Pokedem greatly outperforms competitor Twitter accounts in the same domain in terms of conversion and account growth rates. These results originate from ongoing work in investigating forms of social media automation for supporting account managers in their daily activities.

For the future we plan to further exploit SMT and SocialLink within Pokedem, in all its capabilities as listed in section 7.3.2. Concerning recommendations, we plan to inject DBpedia data obtained via SocialLink links to provide additional features when evaluating a tweet in Pokedem content-based recommender system. Concerning profiling, we plan to use techniques like the ones described in Chapter 6, but this time aimed at categorizing tweets rather than users interests, which can be achieved by placing tweets in the social graph (e.g., via the users involved in the tweet) and then propagating interest categories from popular users in the graph to the target tweet. Concerning analytics, we plan to leverage SMT and SocialLink to automatically discover and suggest relevant accounts of influencers, competitors and other benchmark users to compare against, these accounts found by querying the KB and then mapping the entities found to their Twitter accounts via SocialLink links.

## 7.4 Conclusions

In this chapter, we described the Social Media Toolkit system – the set of tools that facilitate the usage of SocialLink pipeline and resource by providing additional functionality on top of it. The core motivation for building SMT was to realize some of the knowledge transfer scenarios we envisioned while developing SocialLink to prove that knowledge transfer is in fact possible and able to provide meaningful improvements to a variety of tasks. Now it has become an invaluable tool allowing other researchers to build their systems on top of SocialLink much easier and helping to debug and test different setups of the SocialLink family of approaches. We showcase this ability by describing two systems that were developed recently and exploit SMT capabilities: MicroNeel and Pokedem. MicroNeel system was the first one to benefit from the knowledge transfer provided by

SocialLink, acting as a testbed for developing early versions of the project. Pokedem, by contrast, is a much more sophisticated system that was able to embed SocialLink pipeline as a cornerstone for its tweet-enriching functionality. Overall, Social Media Toolkit and the two described systems allowed us to develop SocialLink faster and in a more robust manner by directly observing the effect of various additions to the linking approach on downstream tasks, such as Named Entity Linking.

# Chapter 8

# Conclusions

In this thesis, I introduce the task of linking entities in the LOD cloud to social media profiles aiming to enable knowledge transfer between them. Knowledge transfer here is the transfer of data from one medium to another facilitating a wide variety of tasks at the recipient side. The usefulness of such transfer has been repeatedly shown throughout this thesis. However, as the currently available number of links between the LOD and social media is insignificant compared to the potential amount of entities that could be linked to the profiles, a novel approach was required to populate additional links to bridge the two worlds.

To this end, I presented the SocialLink project that aims at the automatic generation of such links as a core contribution of this thesis. SocialLink is designed to target DBpedia, the cornerstone dataset in the LOD, and Twitter, one of the largest social media. However, even though the scope of the approach presented in this thesis is limited to those two particular data sources, particular attention was given to select the features that would most likely be available in other LOD datasets and social networks, thus enabling future extensions. In this chapter, I will briefly summarize the contributions of this thesis, present some of the possible future work directions and discuss privacy concerns that inevitably arise when working with real people's data.

## 8.1 Summary of Contributions

As presented in Chapter 1, in this thesis, I highlight five main contributions. In this section, I will go over each one of them and briefly summarize the achieved results.

**Contribution 1: SocialLink approach.** One of the major parts of the SocialLink project is the supervised deep learning-based linking approach that relies on the three-phase procedure described in Chapter 3 to produce links between DBpedia and Twitter. Many challenges had to be addressed in order to develop such an approach. In particular, two of them required a significant amount of novelty to be introduced in order to be

resolved. Firstly, the proprietary nature of Twitter meant that some of the aspects of the approach, such as building the social graph embeddings and searching candidate alignments among hundreds of millions of Twitter users, had to be approximated from the available data. Secondly, Twitter's unstructured and uncurated nature meant that the proposed approach had to take into consideration the presence of fake and fan accounts and correctly work with partially missing features. All these challenges were explored in details in Chapter 3.

Additionally, the introduction of the graph-based features from DBpedia and Twitter required the development of a custom topology for the neural network. Efficiently combining two different vector spaces trained from two independent, unsupervised datasets along with regular numeric and categorical features from our BASE feature set required a much more robust approach. Indeed, the baseline solution based on simple concatenation yielded unreliable convergence and offered less performance.

Finally, the SocialLink approach is extensively evaluated, and error analysis is presented to clearly establish the current performance in this task and highlight strengths and weaknesses. The approach is fully reproducible with the complete source code and documentation available online.[1] The two major revisions of the SocialLink pipeline were presented to the research community as two successive publications in proceedings of ACM SAC conference (Nechaev et al., 2017b) and the Progress in Artificial Intelligence journal (Nechaev et al., 2018b).

**Contribution 2: SocialLink resource.** The second part of the SocialLink project is the LOD compliant resource that was produced using the presented approach. To this end, in Chapter 4, I discuss the process of building such dataset by taking 2.5M entities of living people and organizations from 120 DBpedia language chapters and linking them to the corresponding Twitter accounts. The final version of the resource, $v3.0$, proposes candidates for over 1M entities and provides high quality (90% precision) links to 322K of them, significantly increasing the number of available links to Twitter for the entities in the Linked Open Data cloud. Chapter 4 contains detailed statistics, design considerations, filtering and preprocessing techniques; it discusses sustainability and presents some of the most important use cases for such a resource both in the field of Semantic Web and Social Media Analysis. The fact that such a resource can be built at scale of millions of links with just conventional hardware proves that the SocialLink approach is efficient and practical. SocialLink dataset is a part of the LOD cloud,[2] available in multiple different formats for download, and a public SPARQL endpoint is maintained to ease access.

[1] http://w3id.org/sociallink
[2] https://lod-cloud.net/dataset/SocialLink

The rapid increase in link coverage between DBpedia and Twitter is a primary prerequisite for the successful knowledge transfer between the two mediums. Having successfully solved this issue, I continued my study of this topic by aiming at proving that such knowledge transfer in both directions is useful in real-world tasks. Specifically, I addressed in details three applications of the SocialLink resource for *Type Prediction*, *User Profiling* and *Named Entity Linking* that form the rest of the contributions of this thesis.

**Contribution 3: Type Prediction.** To showcase the ability of social media data to benefit Semantic Web community, I set up the pipeline for type prediction on DBpedia. Type prediction, which is the task of predicting missing types for entities in a knowledge base, is typically done by exploiting existing features derived from the knowledge graph of a target KB. Additionally, for such interconnected resources, such as DBpedia, additional resources may be utilized, for example, Wikipedia that has a guaranteed article for each DBpedia entity. In Chapter 5, I defined four feature families (profile, authored text, mentioned text, and social graph) that can be extracted from the Twitter Streaming API and that can be related to the DBpedia entities whose types are predicted through the links of SocialLink. I studied the impact of these additional features enabled by SocialLink, and I have found that in many cases social features allow achieving superior performance compared to the knowledge graph features. Additionally, I studied different combinations of such feature sources, also considering the state-of-the-art Wikipedia-based features as an extra source, finding that the addition of social features provides performance benefits in every case. Such results clearly show that social media data can be used complementary to conventional data sources to improve the type prediction pipelines. Finally, despite the usage of a simple classifier, the combination of all three data sources exhibited great performance levels (e.g., 92% $F_1$ for Location attribute), which suggests its potential for Ontology Population. To summarize, bringing the social data along the links between the social media and the LOD allows greater performance for the LOD-based task of type prediction.

**Contribution 4: Concealing User Interests.** Tasks in social media can benefit from the knowledge transfer in the opposite direction: the ingestion of data from the LOD can enable simplified user profiling pipelines. In Chapter 6, I have described a system that is able to identify user interests in an unsupervised manner. Then, I have presented a novel approach that uses the SocialLink resource to propose a set of actions for the user to perform in order to conceal their digital identity. There, the SocialLink resource provides a set of possible profiles with known interests distribution. Then, the novel approach is tasked to find the ideal configuration of profiles to follow in order to confuse profiling pipelines and make them abstain or infer incorrect interest information about the user.

The proposed concealing approach does not degrade user experience as the additional followed profiles can be filtered out on the user side. While the user profiling pipelines relying on the usage of external resources (Piao and Breslin, 2018) were used before, SocialLink uniquely allows the large scale import of knowledge from DBpedia which made the optimization problem proposed in Chapter 6 possible to solve.

**Contribution 5: Social Media Toolkit.** Named Entity Linking (NEL) is the final task addressed in this thesis exploiting the knowledge transfer. Social Media Toolkit (SMT), presented in Chapter 7, implements two novel NEL scenarios: direct disambiguation of profile mentions found in tweets against DBpedia using the reverse query of SocialLink resource; and the linking of named entities found in a given text to social media profiles employing the custom instance of the SocialLink pipeline. Both of those implementations were embedded into two systems, MicroNeel (Corcoglioniti et al., 2016) and Pokedem (Corcoglioniti et al., 2017, 2018), improving their performances in the respective tasks. Additionally, SMT allows additional customized deployments of the SocialLink pipeline and resource allowing contributors to iterate on the SocialLink project.

## 8.2 Future Work

This thesis covers multiple research topics spread across its five main contributions. A wide variety of extensions and improvements can be made to the approaches and systems presented in Chapters 3-7. In this section, I propose some of the directions that can be explored in future to provide significant impact on top of the work presented in this thesis.

Firstly, SocialLink pipeline and approach may continue to be gradually updated. A significant improvement would consist in the expansion of the pipeline to other social networks, such as Facebook and Instagram, by generalizing the approach used for Twitter making SocialLink even more of a bridge between the social media world and the LOD cloud. By introducing more social media to SocialLink, the approach can include additional measures to ensure that cross-network information is consistent, improving the precision of the overall system. Another critical direction that could be explored is alleviating the recall drop observed during the *candidate acquisition* phase, for example, by redesigning this phase using machine learning-based techniques. Current *candidate selection* approach can also be improved by learning joint embeddings for both social media profiles and knowledge base entities placing them into the same vector space. Finally, the pairwise candidate selection solution could be reformulated to account for all candidate profiles at once, for example, via learning to rank instead of binary classification.

Additional KBs can also be explored. As part of the `soweego` project, which is a project supported by the Wikimedia Foundation, it is planned to extend SocialLink to

specifically target Wikidata entities instead of DBpedia. The goal of `soweego` is to provide references to claims in Wikipedia across the newly populated links, implementing this way another example of real-world knowledge transfer between social media and knowledge bases.

Secondly, concerning the type prediction system from Chapter 5, Wikidata can be used both as the target knowledge graph and as an extra source of links. The usage of the SocialLink resource can help to bolster coverage even further. These additional links will provide (i) more linked entities whose types (where missing) can be predicted and populated using our approach; and (ii) additional training (where types are known), which in turn may allow targeting a more extensive range of types and performing additional analyses, e.g., of the impact on performances of the amount of available social information. Finally, a joint embedding derived from all user-related data can be learned to address privacy concerns when making the data from social media available to the community. Such embeddings can be released as a LOD dataset allowing researchers to seamlessly use social media data in their type prediction and ontology population pipelines.

In third, the concealing approach, described in Chapter 6, can be extended further by covering other user profiling scenarios that target different attributes (e.g. location) and profiling use cases. The concealing approach itself can be improved making it able to learn from the output of the given interests inference pipeline to provide better results on real-world black box systems, such as Twitter's Who To Follow box.

Finally, the SMT, detailed in Chapter 7, may be improved alongside the SocialLink pipeline and approach to support future releases. The UI component can be improved significantly to cover the entirety of the API functionality and additional downstream tasks can be implemented to assist in testing of the pipeline. Greater variety of downstream tasks can help find and resolve corner cases in SocialLink making it more robust and useful. For example, recent works in emoji prediction (Coman et al., 2018) suggested the usage of user-based features, including the ones coming from DBpedia, to improve classification performance. Pipeline-wise, possible improvement directions include replacing Wikimachine with Tint 2.0 (Aprosio and Moretti, 2018) as a default option for Italian and extend SMT to support other languages.

## 8.3 Privacy

The SocialLink project is based on processing terabytes of data about hundreds of millions of people on Twitter. When designing, implementing and releasing any of the resources and approaches based on such data, privacy of those individuals have to be taken into account. In this section, I will address some of the privacy aspects that have to be taken into account when working on or with SocialLink and related approaches. Throughout this

thesis, I employ only the publicly available data extracted from Twitter Streaming API. This API provides the same random subset of tweets for all clients connected to it and is routinely used by researchers and companies for many social media analysis, general NLP and other tasks. Subsets of this data have been released as part of the various datasets before.

Even though the release of the raw Twitter data (or at least a portion of it) would benefit the reproducibility and potentially bolster the adoption of both SocialLink and other works described in this thesis, I believe that it wouldn't be appropriate to do so. Once published, there will be no way for the owners of data (i.e., users) to remove, modify or otherwise control the dissemination of it. For this reason, it would have been irresponsible in my opinion to needlessly decrease the ability of users to control their data even if there is a possibility that some other company or researcher may eventually expose the same information publicly anyway. Additionally, such data release may violate Twitter API's terms of use,[3] which also address the user's right to control the dissemination of their data.

Therefore, the SocialLink resource only releases the internal Twitter user identifier and public user handler without releasing any of the profile information or other user data used during linking, such as the estimated social graph or textual content posted by the user. This is a bare minimum of data that is needed to actually establish the link: the public user handler is needed to fill the existing properties in the ontology such as dbo:isPrimaryTopicOf or foaf:account, while the release of the internal Twitter identifier protects the link in case the handler was changed. For the same reason, when releasing data about our type prediction approach,[4] I omit social media features. Such features can be recomputed using the provided code given sufficiently large sample of Twitter Streaming API.

During the discussion around the soweego[5] project, which aims at embedding the SocialLink pipeline into Wikidata, it became apparent that given the particular use case for linking, and the policies and notability requirements of Wikipedia and related initiatives, even stricter privacy requirements have to be put in place. To this end, the linking between Wikidata and Twitter as part of the project was restricted to "verified" profiles to only include people that were willingly publishing their content online in their official (professional or otherwise) capacity.

As mentioned before, SocialLink enables knowledge transfer between the LOD and social media. In Chapter 4, I have argued that the SocialLink resource is able to make user profiling pipelines and other potentially privacy-intrusive tasks in social media analysis much easier to implement. I confirmed this idea by implementing the interests inference

[3] https://developer.twitter.com/en/developer-terms/policy
[4] https://w3id.org/sociallink/type-prediction
[5] https://meta.wikimedia.org/wiki/Grants:Project/Hjfocs/soweego

pipeline targeted at passive users in Chapter 6. However, as shown in the same chapter, SocialLink can also be used as a cornerstone of the novel approaches aimed at protecting users from inferring their private identity. I urge researchers that would like to build systems on top of SocialLink to always consider the privacy impact of their approaches and develop privacy protection mechanisms instead of the privacy breaching ones.

# Bibliography


Abel, F., Gao, Q., Houben, G., and Tao, K. (2011). Analyzing user modeling on twitter for personalized news recommendations. In *User Modeling, Adaption and Personalization - 19th International Conference, UMAP 2011, Girona, Spain, July 11-15, 2011. Proceedings*, pages 1–12.

Alexander, K., Cyganiak, R., Hausenblas, M., and Zhao, J. (2009). Describing linked datasets. In *Proceedings of the WWW2009 Workshop on Linked Data on the Web, LDOW 2009, Madrid, Spain, April 20, 2009.*

Aprosio, A. P., Giuliano, C., and Lavelli, A. (2013). Automatic Expansion of DBpedia Exploiting Wikipedia Cross-Language Information. In *The Semantic Web: Semantics and Big Data, 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings*, volume 7882 of *Lecture Notes in Computer Science*, pages 397–411. Springer.

Aprosio, A. P. and Moretti, G. (2018). Tint 2.0: An all-inclusive suite for nlp in italian. *Proceedings of CLIC-it.*

Armentano, M. G., Godoy, D., and Amandi, A. (2012). Topology-Based Recommendation of Users in Micro-Blogging Communities. *Journal of Computer Science and Technology*, 27(3):624–634.

Armentano, M. G., Godoy, D., and Amandi, A. A. (2013). Followee recommendation based on text analysis of micro-blogging activity. *Information Systems*, 38(8):1116–1127.

Barbieri, N., Bonchi, F., and Manco, G. (2014). Who to Follow and Why: Link Prediction with Explanations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1266–1275, New York, NY, USA. ACM.

Barbu, O. (2014). Advertising, Microtargeting and Social Media. *Procedia - Social and Behavioral Sciences*, 163:44–49.

Basile, P., Caputo, A., Gentile, A. L., and Rizzo, G. (2016a). Overview of the EVALITA 2016 Named Entity rEcognition and Linking in Italian tweets (NEEL-IT) task. In *Proceedings of the 5th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA 2016)*. aAcademia University Press.

Basile, P., Cutugno, F., Nissim, M., Patti, V., and Sprugnoli, R. (2016b). EVALITA 2016: Overview of the 5th evaluation campaign of natural language processing and speech tools for Italian. In *Proceedings of the 5th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA 2016)*. aAcademia University Press.

Bengio, Y. (2013). Estimating or propagating gradients through stochastic neurons. *CoRR*, abs/1305.2982.

Berkhin, P. (2006). Bookmark-coloring approach to personalized pagerank computing. *Internet Mathematics*, 3(1):41–62.

Berners-Lee, T. (2006). Linked Data — Design Issues. https://www.w3.org/DesignIssues/LinkedData.html. [Online; accessed 26-Dec-2018].

Besel, C., Schlötterer, J., and Granitzer, M. (2016). Inferring Semantic Interest Profiles from Twitter Followees: Does Twitter Know Better Than Your Friends? In *ACM SAC*, pages 1152–1157.

Bettini, C. and Riboni, D. (2015). Privacy protection in pervasive systems: State of the art and technical challenges. *Pervasive and Mobile Computing*, 17(Part B):159 – 174. 10 years of Pervasive Computing' In Honor of Chatschik Bisdikian.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M. A., Maynard, D., and Aswani, N. (2013). TwitIE: An open-source information extraction pipeline for microblog text. In *Recent Advances in Natural Language Processing, RANLP*, pages 83–90. RANLP 2013 Organising Committee / ACL.

Bordes, A., Usunier, N., Garc\'\ia-Durán, A., Weston, J., and Yakhnenko, O. (2013). Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013*, pages 2787–2795.

Brunton, F. and Nissenbaum, H. (2015). *Obfuscation: A user's guide for privacy and protest.* The MIT Press.

Cai, H., Zheng, V. W., and Chang, K. C. (2018). A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Trans. Knowl. Data Eng.*, 30(9):1616–1637.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27.

Chaoji, V., Ranu, S., Rastogi, R., and Bhatt, R. (2012). Recommendations to Boost Content Spread in Social Networks. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 529–538, New York, NY, USA. ACM.

Church, K. W. and Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Cochez, M., Ristoski, P., Ponzetto, S. P., and Paulheim, H. (2017a). Biased graph walks for RDF graph embeddings. In *Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics, WIMS 2017*, pages 21:1—-21:12.

Cochez, M., Ristoski, P., Ponzetto, S. P., and Paulheim, H. (2017b). Global RDF Vector Space Embeddings. In *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part I*, volume 10587 of *Lecture Notes in Computer Science*, pages 190–207. Springer.

Coman, A. C., Nechaev, Y., and Zara, G. (2018). Predicting Emoji Exploiting Multimodal Data: FBK Participation in ITAmoji Task. In *Proceedings of Fifth Italian Conference on Computational Linguistics (CLiC-it 2018) & Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018).*

Corcoglioniti, F., Aprosio, A. P., Nechaev, Y., and Giuliano, C. (2016). MicroNeel: Combining NLP Tools to Perform Named Entity Detection and Linking on Microposts. In *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016), Napoli, Italy, December 5-7, 2016.*

Corcoglioniti, F., Giuliano, C., Nechaev, Y., and Zanoli, R. (2017). Pokedem: An Automatic Social Media Management Application. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, pages 358–359, New York, NY, USA. ACM.

Corcoglioniti, F., Nechaev, Y., Giuliano, C., and Zanoli, R. (2018). Twitter User Recommendation for Gaining Followers. In *AI*IA 2018 Advances in Artificial Intelligence - 17th International Conference of the Italian Association for Artificial Intelligence, Trento, Italy, November 20-23, 2018, Proceedings.*

Corcoglioniti, F., Rospocher, M., Mostarda, M., and Amadori, M. (2015). Processing billions of {RDF} triples on a single machine using streaming and sorting. In *ACM SAC*, pages 368–375.

Cresci, S., Pietro, R. D., Petrocchi, M., Spognardi, A., and Tesconi, M. (2015). Fame for sale: Efficient detection of fake Twitter followers. *Decision Support Systems*, 80:56–71.

Cristianini, N., Shawe-Taylor, J., and Lodhi, H. (2002). Latent Semantic Kernels. *Journal of Intelligent Information Systems*, 18(2):127–152.

de Vries, G. K. D. (2013). A fast approximation of the weisfeiler-lehman graph kernel for RDF data. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part I*, pages 606–621.

de Vries, G. K. D. and de Rooij, S. (2015). Substructure counting graph kernels for machine learning from RDF data. *J. Web Sem.*, 35:71–84.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). Indexing by latent semantic analysis. *JASIS*, 41(6):391–407.

Dwork, C. (2008). Differential privacy: A survey of results. In *Proc. of 5th Int. Conf. on Theory and Applications of Models of Computation (TAMC)*, pages 1–19, Berlin, Heidelberg. Springer-Verlag.

Erxleben, F., Günther, M., Krötzsch, M., Mendez, J., and Vrande\uaić, D. (2014). Introducing Wikidata to the Linked Data Web. In *Proceedings of the 13th International Semantic Web Conference - Part I*, ISWC '14, pages 50–65, New York, NY, USA. Springer-Verlag New York, Inc.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874.

Faralli, S., Stilo, G., and Velardi, P. (2015a). Large Scale Homophily Analysis in Twitter Using a Twixonomy. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015*, pages 2334–2340.

Faralli, S., Stilo, G., and Velardi, P. (2015b). Recommendation of microblog users based on hierarchical interest profiles. *Social Network Analysis and Mining*, 5(1):25.

Farseev, A., Nie, L., Akbari, M., and Chua, T.-S. (2015). Harvesting Multiple Sources for User Profile Learning: A Big Data Study. In *ACM ICMR*, pages 235–242.

Felt, A. and Evans, D. (2008). Privacy protection for social networking APIs.

Fetahu, B., Anand, A., and Anand, A. (2015). How Much is Wikipedia Lagging Behind News? In *Proceedings of the ACM Web Science Conference*, WebSci '15, pages 28:1—-28:9, New York, NY, USA. ACM.

Goga, O. (2014). *Matching user accounts across online social networks: methods and applications*. PhD thesis, LIP6-Laboratoire d'Informatique de Paris 6.

Goga, O., Lei, H., Parthasarathi, S. H. K., Friedland, G., Sommer, R., and Teixeira, R. (2013). Exploiting Innocuous Activity for Correlating Users Across Sites. In *Proc. of WWW*, pages 447–458. ACM.

Goga, O., Loiseau, P., Sommer, R., Teixeira, R., and Gummadi, K. P. (2015). On the Reliability of Profile Matching Across Large Online Social Networks. In *Proc. of KDD*, pages 1799–1808. ACM.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Goyal, P. and Ferrara, E. (2017). Graph embedding techniques, applications, and performance: A survey. *arXiv preprint arXiv:1705.02801*.

Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *The 22th {ACM} {SIGKDD} International Conference on Knowledge Discovery and Data Mining, {KDD} '16*, pages 855–864. ACM.

Guha, R. and Brickley, D. (2014). RDF Schema 1.1. W3C Recommendation, W3C. http://www.w3.org/TR/2014/REC-rdf-schema-20140225/.

Gupta, P., Goel, A., Lin, J., Sharma, A., Wang, D., and Zadeh, R. (2013). WTF: The Who to Follow Service at Twitter. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 505–514, New York, NY, USA. ACM.

Guu, K., Miller, J., and Liang, P. (2015). Traversing knowledge graphs in vector space. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 318–327.

Hannon, J., Bennett, M., and Smyth, B. (2010). Recommending Twitter Users to Follow Using Content and Collaborative Filtering Approaches. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 199–206, New York, NY, USA. ACM.

Harris, S. and Seaborne, A. (2013). SPARQL 1.1 query language. W3C Recommendation, W3C. http://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

Hoffart, J., Suchanek, F. M., Berberich, K., and Weikum, G. (2013). YAGO2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61.

Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., and Bengio, Y. (2016). Binarized Neural Networks. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 4107–4115. Curran Associates, Inc.

Kejriwal, M. and Szekely, P. (2017). Supervised typing of big graphs using semantic embeddings. In *Proceedings of The International Workshop on Semantic Big Data, SBD@SIGMOD 2017, Chicago, IL, USA, May 19, 2017*, pages 3:1—-3:6.

Kim, Y. and Shim, K. (2014). TWILITE: A recommendation system for Twitter using a probabilistic model based on latent Dirichlet allocation. *Information Systems*, 42:59–77.

Kywe, S. M., Lim, E.-P., and Zhu, F. (2012). A Survey of Recommender Systems in Twitter. In *Proceedings of the 4th International Conference on Social Informatics*, SocInfo'12, pages 420–433, Berlin, Heidelberg. Springer-Verlag.

Landauer, T. K., Foltz, P. W., and Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., and Bizer, C. (2015). DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6(2):167–195.

Li, J., Ritter, A., and Hovy, E. (2014). Weakly Supervised User Profile Extraction from Twitter. In *Proc. of 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 165–174.

Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 2181–2187.

Liu, S., Wang, S., Zhu, F., Zhang, J., and Krishnan, R. (2014). HYDRA: Large-scale social identity linkage via heterogeneous behavior modeling. In *Proc. of SIGMOD*, pages 51–62. ACM.

Lops, P., de Gemmis, M., and Semeraro, G. (2011). Content-based Recommender Systems: State of the Art and Trends. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 73–105. Springer US, Boston, MA.

Lu, C.-T., Shuai, H.-H., and Yu, P. S. (2014). Identifying Your Customers in Social Networks. In *Proc. of CIKM*, pages 391–400. ACM.

Ludlow, K. (2012). Bayesian flooding and Facebook manipulation.

Lund, K. and Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208.

Luo, W., Xie, Q., and Hengartner, U. (2009). FaceCloak: An architecture for user privacy on social networking sites. In *Proc. of Int. Conf. on Computational Science and Engineering*, volume 3, pages 26–33.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*, pages 55–60.

Melo, A., Paulheim, H., and Völker, J. (2016). Type Prediction in RDF Knowledge Bases Using Hierarchical Multilabel Classification. In *Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics*, WIMS '16, pages 14:1—-14:10.

Michelson, M. and Macskassy, S. A. (2010). Discovering users' topics of interest on twitter: a first look. In *Proceedings of the Fourth Workshop on Analytics for Noisy Unstructured Text Data, AND 2010, Toronto, Ontario, Canada, October 26th, 2010 (in conjunction with CIKM 2010)*, pages 73–80.

Minard, A., Qwaider, M. R. H., and Magnini, B. (2016). FBK-NLP at NEEL-IT: Active Learning for Domain Adaptation. In *EVALITA*.

Mislove, A., Viswanath, B., Gummadi, P. K., and Druschel, P. (2010). You are who you know: inferring user profiles in online social networks. In *Proceedings of the Third International Conference on Web*

*Search and Web Data Mining, WSDM 2010, New York, NY, USA, February 4-6, 2010*, pages 251–260.

Motik, B., Patel-Schneider, P., and Parsia, B. (2012). OWL 2 web ontology language structural specification and functional-style syntax (second edition). W3C recommendation, W3C. http://www.w3.org/TR/2012/REC-owl2-syntax-20121211/.

Nechaev, Y., Corcoglioniti, F., and Giuliano, C. (2017a). Concealing Interests of Passive Users in Social Media. In *Proceedings of the Re-coding Black Mirror 2017 Workshop co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 22, 2017*.

Nechaev, Y., Corcoglioniti, F., and Giuliano, C. (2017b). Linking Knowledge Bases to Social Media Profiles. In *Proceedings of the Symposium on Applied Computing, SAC 2017, Marrakech, Morocco, April 3-7, 2017*, pages 145–150.

Nechaev, Y., Corcoglioniti, F., and Giuliano, C. (2017c). SocialLink: knowledge transfer between social media and linked open data. https://doi.org/10.6084/m9.figshare.5235823.

Nechaev, Y., Corcoglioniti, F., and Giuliano, C. (2017d). SocialLink: Linking DBpedia Entities to Corresponding Twitter Accounts. In *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II*, pages 165–174.

Nechaev, Y., Corcoglioniti, F., and Giuliano, C. (2018a). SocialLink dataset v3.0. https://doi.org/10.5281/zenodo.1451797.

Nechaev, Y., Corcoglioniti, F., and Giuliano, C. (2018b). SocialLink: Exploiting Graph Embeddings to Link DBpedia Entities to Twitter Profiles. *Progress in AI*, 7(4):251–272.

Nechaev, Y., Corcoglioniti, F., and Giuliano, C. (2018c). Type Prediction Combining Linked Open Data and Social Media. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 1033–1042.

Nickel, M., Rosasco, L., and Poggio, T. A. (2016). Holographic embeddings of knowledge graphs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 1955–1961.

Nickel, M., Tresp, V., and Kriegel, H. (2011). A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 809–816.

Noreen, E. W. (1989). *Computer-intensive methods for testing hypotheses*. Wiley New York.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab. Previous number = SIDL-WP-1999-0120.

Palmero Aprosio, A. and Giuliano, C. (2016). The wiki machine: an open source software for entity linking and enrichment. *ArXiv e-prints*.

Palmero Aprosio, A. and Moretti, G. (2016). Italy goes to Stanford: a collection of CoreNLP modules for Italian. *ArXiv e-prints*.

Paniagua, J. and Sapena, J. (2014). Business performance and social media: Love or hate? *Business Horizons*, 57(6):719–728.

Paulheim, H. and Bizer, C. (2013). Type Inference on Noisy {RDF} Data. In *The Semantic Web - {ISWC} 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part {I}*, pages 510–525.

Peled, O., Fire, Rokach, and Elovici (2016). Matching entities across online social networks. *Neurocomputing*.

Pennington, J., Socher, R., and Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). DeepWalk: online learning of social representations. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*, pages 701–710.

Piao, G. and Breslin, J. G. (2016). 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, Leipzig, Germany, September 12-15, 2016*, pages 81–88.

Piao, G. and Breslin, J. G. (2017). Inferring User Interests for Passive Users on Twitter by Leveraging Followee Biographies. In *Advances in Information Retrieval - 39th European Conference on {IR} Research, {ECIR} 2017*, pages 122–133.

Piao, G. and Breslin, J. G. (2018). Inferring user interests in microblogging social networks: a survey. *User Model. User-Adapt. Interact.*, 28(3):277–329.

Preotiuc-Pietro, D., Lampos, V., and Aletras, N. (2015). An analysis of the user occupational class through Twitter content. In *Proceedings of 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1754–1764. The Association for Computer Linguistics.

Rao, D., Yarowsky, D., Shreevats, A., and Gupta, M. (2010). Classifying latent user attributes in Twitter. In *Proceedings of 2nd International Workshop on Search and Mining User-generated Contents (SMUC)*, pages 37–44.

Rico, M., Santana-Perez, I., Pozo-Jimenez, P., and Gomez-Perez, A. (2018). Inferring New Types on Large Datasets Applying Ontology Class Hierarchy Classifiers: The DBpedia Case. In *Proceedings of the 15th Extended Semantic Web Conference (ESWC)*.

Ristoski, P. and Paulheim, H. (2016a). Rdf2vec: RDF graph embeddings for data mining. In *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part I*, pages 498–514.

Ristoski, P. and Paulheim, H. (2016b). Semantic Web in data mining and knowledge discovery: A comprehensive survey. *Web Semantics: Science, Services and Agents on the World Wide Web*, 36:1–22.

Ristoski, P., Rosati, J., Di Noia, T., De Leone, R., and Paulheim, H. (2017). RDF2Vec: RDF graph embeddings and their applications. *Semantic Web Journal*.

Sadilek, A., Kautz, H., and Bigham, J. P. (2012). Finding Your Friends and Following Them to Where You Are. In *Proc. of 5th ACM Int. Conf. on Web Search and Data Mining (WSDM)*, pages 723–732, New York, NY, USA. ACM.

Shazeer, N., Doherty, R., Evans, C., and Waterson, C. (2016). Swivel: Improving Embeddings by Noticing What's Missing. *CoRR*, abs/1602.02215.

Siehndel, P. and Kawase, R. (2012). Twikime! - user profiles that make sense. In *Proceedings of the ISWC 2012 Posters & Demonstrations Track, Boston, USA, November 11-15, 2012*.

Socher, R., Chen, D., Manning, C. D., and Ng, A. Y. (2013). Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 926–934.

statista.com (2019a). Facebook users worldwide 2018. https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/. [Online; accessed 15-Jan-2019].

statista.com (2019b). Instagram: active users 2018. https://www.statista.com/statistics/253577/number-of-monthly-active-instagram-users/. [Online; accessed 15-Jan-2019].

statista.com (2019c). Twitter: number of active users 2010-2018. https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/. [Online; accessed 15-Jan-2019].

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. (2015). LINE: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077.

Tommasel, A., Corbellini, A., Godoy, D., and Schiaffino, S. (2016). Personality-aware followee recommendation algorithms: An empirical analysis. *Engineering Applications of Artificial Intelligence*, 51:24–36.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014). Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pages 1112–1119.

Wilkinson, M. D. et al. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*.

Wood, D., Lanthaler, M., and Cyganiak, R. (2014). RDF 1.1 concepts and abstract syntax. W3C Recommendation, W3C. http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/.

Yanardag, P. and Vishwanathan, S. V. N. (2015). Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pages 1365–1374.

Yuan, G., Murukannaiah, P. K., Zhang, Z., and Singh, M. P. (2014). Exploiting Sentiment Homophily for Link Prediction. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 17–24, New York, NY, USA. ACM.

Zafarani, R. and Liu, H. (2009). Connecting Corresponding Identities across Communities. In *Proc. of ICWSM*. AAAI Press.

Zafarani, R. and Liu, H. (2013). Connecting Users Across Social Media Sites: A Behavioral-modeling Approach. In *Proc. of KDD*, pages 41–49. ACM.

Zanchetta, E. and Baroni, M. (2005). Morph-it! a free corpus-based morphological resource for the Italian language. *Corpus Linguistics 2005*, 1(1).

Zarrinkalam, F., Fani, H., Bagheri, E., and Kahani, M. (2016). Inferring implicit topical interests on twitter. In *Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings*, pages 479–491.

Zhang, Y., Rahman, M. M., Braylan, A., Dang, B., Chang, H., Kim, H., McNamara, Q., Angert, A., Banner, E., Khetan, V., McDonnell, T., Nguyen, A. T., Xu, D., Wallace, B. C., and Lease, M. (2016). Neural information retrieval: A literature review. *CoRR*, abs/1611.06792.

Zhao, G., Lee, M. L., Hsu, W., Chen, W., and Hu, H. (2013). Community-based User Recommendation in Uni-directional Social Networks. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 189–198, New York, NY, USA. ACM.

Zheleva, E. and Getoor, L. (2009). To Join or Not to Join: The Illusion of Privacy in Social Networks with Mixed Public and Private User Profiles. In *Proc. of 18th Int. Conf. on World Wide Web (WWW)*, pages 531–540, New York, NY, USA. ACM.