
**Fakultät für Informatik und Mathematik**
**Professur für Informatik mit Schwerpunkt Medieninformatik**

**Dissertation**

# Robust Entity Linking in Heterogeneous Domains

M.Sc. Stefan Zwicklbauer

Januar 2017

1. Gutachter: Prof. Dr. Michael Granitzer
2. Gutachter: Prof. Dr. York Sure-Vetter

## Acknowledgments

*Firstly, I would like to express my sincere gratitude to my advisor Prof. Granitzer for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.*

*Besides my advisor, my sincere thanks goes to Dr. Christin Seifert. Thank your for your valuable feedback, continuous support and guidance in the last years and for keeping me motivated throughout the writing of the papers and this thesis. This work would not have been possible without your help and input.*

*I really appreciate the support, sophisticated feedback and friendly advice from my outstanding colleagues during my period at the professorship of media computer science (in alphabetic order): Sebastian Bayerl, Annemarie Gertler-Weber, Johannes Jurgovsky, Prof. Dr. Harald Kosch, Kevin Maurer, Albin Petit, Fatemeh Salehi Rizi, Jörg Schlötterer, Kai Schlegel, Thorben Schnuchel, Andreas Wölfl and Dr. Konstantin Ziegler.*

*Finally, I would like to thank my parents and my girlfriend **Julia** for supporting me throughout writing this thesis and my life in general. Your devotion, unconditional love and support, sense of humour, patience, optimism and advice was more valuable than you could ever imagine.*

## Abstract

Entity Linking is the task of mapping terms in arbitrary documents to entities in a knowledge base by identifying the correct semantic meaning. It is applied in the extraction of structured data in RDF (Resource Description Framework) from textual documents, but equally so in facilitating artificial intelligence applications, such as Semantic Search, Reasoning and Question and Answering. Most existing Entity Linking systems were optimized for specific domains (e.g., general domain, biomedical domain), knowledge base types (e.g., DBpedia, Wikipedia), or document structures (e.g., tables) and types (e.g., news articles, tweets). This led to very specialized systems that lack robustness and are only applicable for very specific tasks. In this regard, this work focuses on the research and development of a robust Entity Linking system in terms of domains, knowledge base types, and document structures and types.

To create a robust Entity Linking system, we first analyze the following three crucial components of an Entity Linking algorithm in terms of robustness criteria: (i) the underlying knowledge base, (ii) the entity relatedness measure, and (iii) the textual context matching technique. Based on the analyzed components, our scientific contributions are three-fold. First, we show that a federated approach leveraging knowledge from various knowledge base types can significantly improve robustness in Entity Linking systems. Second, we propose a new state-of-the-art, robust entity relatedness measure for topical coherence computation based on semantic entity embeddings. Third, we present the neural-network-based approach Doc2Vec as a textual context matching technique for robust Entity Linking.

Based on our previous findings and outcomes, our main contribution in this work is DoSeR (Disambiguation of Semantic Resources). DoSeR is a robust, knowledge-base-agnostic Entity Linking framework that extracts relevant entity information from multiple knowledge bases in a fully automatic way. The integrated algorithm represents a collective, graph-based approach that utilizes semantic entity and document embeddings for entity relatedness and textual context matching computation. Our evaluation shows, that DoSeR achieves state-of-the-art results over a wide range of different document structures (e.g., tables), document types (e.g., news documents) and domains (e.g., general domain, biomedical domain). In this context, DoSeR outperforms all other (publicly available) Entity Linking algorithms on most data sets.



**Keywords:** Entity Linking, Neural Networks, Linked Data, Knowledge Bases

# Contents

# Part I

Preface

# CHAPTER 1

# Introduction

## 1.1 Motivation

The World Wide Web has grown rapidly in the last decade and has become the most important information source for most of us. In its current form, the Web offers a corpus comprising more than 40 billion web pages[1], including 233 million tables according to the WDC Web Table Corpus 2015[2] and a vast number of blogs and social networks with millions of user profiles. Most Web data is unstructured, interconnected, noisy and often expressed in the form of natural language text. This inspired the construction of knowledge bases (KB) which typically contain a wealth of information about real-world entities and how they are linked to each other, including hierarchies, taxonomies and other semantic relations. A set of notable KB examples includes Wikipedia[3], DBpedia [Aue07], YAGO [Suc07], Freebase [Bol08], Probase [Wu12] and DBLP[4]. Bridging unstructured (Web) data and KBs is crucial and beneficial in terms of annotating raw and noisy data as well as contributing to the vision of the Semantic Web [Ber01]. A critical step to achieve this goal is Entity Linking (EL). EL is the task of establishing links between selected text fragments, also known as surface forms, and their correct semantic meaning (an entity represented as a unique ID) from a set of candidate meanings (referred to as the KB). The EL task is challenging due to the ambiguity of many surface forms: a surface form can refer to different entities depending on the respective context provided by the documents. For example, Figure 1.1 depicts a research article extract comprising textual content and a table. Both, the text and the table, contain the surface form 'tree' (yellow highlighted rectangles), which refers to the entity *Tree (data structure)* in the underlying KB (blue highlighted container). EL algorithms, however, aim to resolve the ambiguity of all surface forms located in the article and link them to the respective entities.

EL can greatly facilitate many different tasks such as Knowledge Base Population, Semantic Search and Question and Answering. With data and facts accumulating in the Web over time, the enrichment of existing KBs becomes more and more important. Integrating new knowledge extracted from information extraction systems into KBs demands systems to link surface forms associated with the new extracted facts with the respective entities in the KB. Semantic Search systems aim to improve search accuracy by understanding the

1 http://www.worldwidewebsize.com/, last accessed on 2016-11-29
2 http://webdatacommons.org/webtables/, last accessed on 2016-11-29
3 http://www.wikipedia.org/, last accessed on 2016-11-29
4 http://www.dblp.org/, last accessed on 2016-11-29

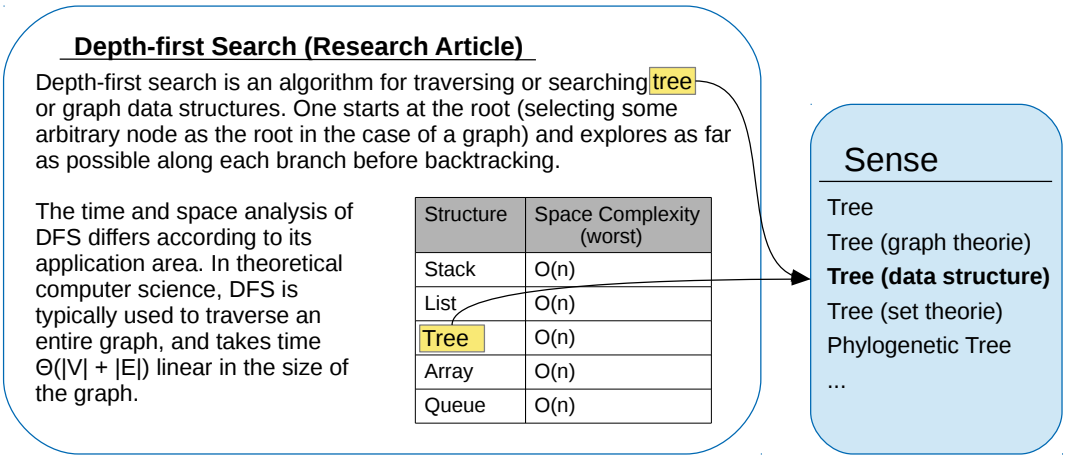



**Figure 1.1:** A research document that contains a surface form in the text and table (yellow highlighted rectangles). EL algorithms aim to resolve the ambiguity of the surface forms and link them to the corresponding entity in a KB (blue highlighted container).

searcher's intent and, thus, rely on EL techniques to understand what a user is searching for. In Question and Answering systems, however, the system leverages an EL algorithm, allowing it to fetch relevant information from the respective KB entry.

Given various tasks and applications, EL has received significant attention from the Natural Language Processing, Semantic Web, Data Mining and Information Retrieval community. As a result, these communities brought up a plethora of different EL frameworks, algorithms and techniques. A careful analysis of related work across different communities and domains reveals some core limitations of most EL systems, such as domain dependency. In practice, EL systems are optimized toward a specific domain, resulting in approaches that are particularly suited to link general-domain entities (e.g., persons, organizations, locations) or special-domain entities (e.g., genes, proteins). Linking entities across multiple domains requires different EL systems because there is a lack of domain-spanning approaches so far. In this context, a related problem occurs when EL systems consider entities from multiple domains or KBs. Covering entities from different domains leads to large and heterogeneous KBs. We assume that this complicates the EL process due to an increase of potential candidate entities per surface form. Moreover, the way entities are described within a KB plays a crucial role. The KB structure is typically not consistent across KBs, resulting in two major KB types: graph-based (e.g., DBpedia) and entity-annotated document KBs (e.g., Wikipedia). KB-agnostic frameworks that consider different types of knowledge are still rare [Usb14] yet necessary to construct highly accurate, robust EL systems. Another limitation is the optimization on specific document structures (e.g., textual documents, tables) and types (e.g., news articles, tweets). While EL systems for disambiguating entities in news articles, research documents or tweets have been partially compared using different data sets, there have been no experiments on table data sets. This led to very specialized systems that lack robustness and perform well exclusively on specific document structures and types.

In **summary**, EL algorithms have been well researched in the literature. However, most existing EL systems are optimized toward specific document structures and types as well as specific domains to achieve superior EL results. Furthermore, EL systems are generally not agnostic in terms of different KB structures, such as graph-based and entity-annotated document KBs.

## 1.2 Research Objectives

Based on the limitations of current EL systems, the ultimate goal and our main research objective in this work is to create a robust EL system in terms of domains, KB types, and document structures and types. To this end, we first define two crucial characteristics that quantify the term *Robustness* in the context of EL systems:

- Structural Robustness, i.e., EL systems are agnostic in terms of different KB types and provide consistent results across various document structures and types.
- Consistency, i.e., EL systems achieve consistent results across various domains, with a low quantity and/or poor quality of entity descriptions as well as on large-scale and heterogeneous KBs.

While quantifying the term *Robustness* in EL systems is the first important step, it remains unclear how to fulfill all robustness criteria in an EL system. For this purpose, we first identify three major components of EL algorithms. Then, for each component, we analyze and propose techniques that satisfy the defined robustness criteria. The main components of EL systems analyzed in this work are: (i) the underlying KB, (ii) the entity relatedness measure, and (iii) the textual context matching technique. Given these components, our additional research objectives are as follows:

1. **Knowledge Bases:** Since the underlying KB plays an important role in EL systems, we question how different KB properties influence EL results. More specifically, we strive to investigate the influence of (i) the entity format (i.e., the way entities are described), (ii) user data (i.e., the quantity and quality of externally disambiguated entities), and (iii) the quantity and quality of the entities to disambiguate.
2. **Entity Relatedness:** Linking multiple surface forms within the same document in one step can significantly improve the EL results. We question which entity relatedness measure achieves state-of-the-art results in EL systems while providing Structural Robustness and Consistency.
3. **Textual Context:** Textual context matching techniques are typically employed in all EL systems. However, their influence on EL results strongly depends on the surrounding context of surface forms and the quantity and quality of entity descriptions in the underlying KBs. We question which textual context matching technique achieves state-of-the-art results in EL systems while providing Structural Robustness and Consistency.

***Methodology***
We use an experimental research methodology throughout this work. After specifying the concept and research questions of this work in Chapter 4, we approach each scientific question by conducting experiments with different variables. These are different types or parts of EL algorithms that are compared using multiple data sets given various conditions. Conditions are chosen according to the respective robustness criteria that are evaluated, for instance, different amounts of (noisy) user data or different degrees of heterogeneity in KBs. All claims and conclusions are drawn from the conducted experiments and their respective result values.

## 1.3 Scientific Contributions

Based on our research objectives defined in Section 1.2, we present the scientific contributions of this work in the following.

**Knowledge Bases:** We provide a systematic evaluation of biomedical EL with respect to the crucial KB properties entity format, user data and quantity and heterogeneity of entities [Zwi13b; Zwi15a; Zwi15c]. In this context, our EL results reveal that the entity format (i.e., graph-based KBs or entity-annotated document KBs) that is used to achieve the best EL results strongly depends on the amount of available user data. Moreover, the entity format strongly affects EL results with large-scale and heterogeneous KBs. Finally, we show that a federated approach leveraging different types of entity definitions (i.e., different entity formats) can significantly improve the Consistency of EL systems.

**Entity Relatedness:** We propose a new state-of-the-art entity relatedness measure for collective EL based on semantic embeddings [Zwi16a]. We create these semantic embeddings with Word2Vec and propose how to automatically generate appropriate Word2Vec input corpora based on different KBs. To evaluate our relatedness measure, we integrated it in a simple yet collective EL approach. Our approach outperforms existing and more complex, publicly available, state-of-the-art approaches on most data sets in our evaluation. We also conducted experiments on different KBs, various document structures and types, and varying quantities and qualities of entity definitions/annotations. Our experiments show that our approach provides Structural Robustness and Consistency.

**Textual Context:** We provide a systematic evaluation of state-of-the-art textual context matching techniques for EL systems with regard to the structure and type of documents and the quantity of entity descriptions. In our evaluation, we compare the neural-network-based approach Doc2Vec to two TF-IDF-based approaches (i.e., Vector Space Model approach with TF-IDF weights and Okapi BM-25), a language model approach (i.e., Entity-Context Model) and a Latent Dirichlet Allocation approach (i.e., Thematic Context Distance). Overall, we show that Doc2Vec provides Structural Robustness and Consistency with short and extensive entity descriptions in the underlying KB.

**DoSeR:** Based on the previous findings and scientific contributions, we present the EL framework DoSeR [Zwi16a; Zwi16b] (**D**isambiguation **o**f **Se**mantic **R**esources). DoSeR is a KB-agnostic framework that achieves state-of-the-art results across different domains

(i.e., linking general-domain and special-domain entities). Moreover, it provides Structural Robustness and Consistency in terms of most criteria on the evaluated data sets and KBs. The underlying DoSeR EL algorithm represents a collective, graph-based approach that utilizes semantic entity (Word2Vec) and document embeddings (Doc2Vec) for robust EL. Our approach is also able to abstain if no appropriate entity can be found for a specific surface form. We provide our framework as well as the underlying KB as an open source solution to allow a fair comparison between future EL systems.

## 1.4 Structure

This work contains an introduction, three major parts consisting of seven chapters, and a conclusion. The first major part describes related work. It gives a detailed overview of all important parts of EL systems, including different EL tasks, KBs and algorithms. In the second and main part of this work, we first describe three core limitations concerning (state-of-the-art) EL approaches. Further, we define the characteristics of Structural Robustness and Consistency for EL systems. We select three important components of EL algorithms, namely the underlying KB, the entity relatedness measure and the textual context matching technique, and investigate them in terms of robust EL. Hereby, we propose respective techniques on how to improve the Robustness within EL algorithms. In the third part, we present DoSeR, a robust, state-of-the-art EL framework that unifies the findings of the previous chapters. In the following, we briefly summarize each chapter:

**Chapter 2** gives an introduction to EL in general. More specifically, we formalize the problem of EL and present related tasks. We also provide a brief overview of popular KBs that have often been used as entity databases and describe evaluation metrics for EL systems.

**Chapter 3** provides an extensive overview of existing EL approaches. We subdivide the chapter into candidate entity generation, EL features, disambiguation algorithms and abstaining. Candidate entity generation describes techniques that select relevant entities for each surface form. In the EL features section, we provide an overview of typical features used to compute a relevance score of how well a candidate entity fits to its respective surface form. In the next section, we describe current state-of-the-art methods to model and find the most appropriate candidate entity for each surface form given an input document. Finally, we briefly discuss abstaining methods to detect unlinkable surface forms.

**Chapter 4** analyzes three crucial shortcomings in related work, namely domain dependency, KB properties, and document structures and types. Based on these shortcomings, we define the term *Robustness* for EL systems as an umbrella term that covers two crucial characteristics for EL systems: Structural Robustness and Consistency. In this context, we pose our main research question, which addresses the construction of a robust EL system. To create such a system, we identify three crucial components of EL algorithms that contribute to robust EL (i.e., the underlying KB, the entity relatedness measure and the textual context matching technique). For each component, we discuss issues in terms of Robustness and pose further research questions. We also provide a brief outlook of the subsequent chapters and summarize the respective outcomes.

**Chapter 5** investigates how KB properties influence EL results. To this end, we define and model the KB properties (i) entity format, i.e., the way entities are described, (ii) user data, i.e., the quantity and quality of externally disambiguated entities, and (iii) the quantity and heterogeneity of entities, i.e., the number and size of different domains in a KB. We implemented three ranking-based EL algorithms to address different types of KBs and investigate how and to what degree the defined KB properties influence EL results in terms of Consistency.

**Chapter 6** presents a new state-of-the-art entity relatedness measure based on entity embeddings (Word2Vec) for collective EL. Moreover, we show how to easily generate these entity embeddings to compute semantic similarities between entities regardless of the underlying KB type (graph-based or entity-annotated document KBs). We conducted experiments to show that our new measure achieves state-of-the-art results while providing Structural Robustness and Consistency.

**Chapter 7** investigates which textual context matching technique provides Structural Robustness and Consistency while providing state-of-the-art results in EL approaches. Overall, we analyze two TF-IDF-based approaches (i.e., Vector Space Model approach and Okapi BM-25), a language model approach, a Latent Dirichlet Allocation approach and a neural-network-based approach (i.e., Doc2Vec) in a systematic evaluation. We implemented a feature-reduced (evaluating textual context matching techniques only) EL algorithm to isolate and evaluate the textual context matching techniques. We conducted experiments on different KBs to analyze how the approaches perform on various document structures and types.

**Chapter 8** describes DoSeR, a (named) EL framework that is KB-agnostic in terms of graph-based (e.g., DBpedia) and entity-annotated document KBs (e.g., Wikipedia). We first describe how our framework automatically generates an EL index given one or multiple KBs to store necessary entity information for our approach later on. Further, we propose a new collective, graph-based EL algorithm that integrates our robust entity relatedness measure for topical coherence computation (Word2Vec) and the robust textual context matching technique (Doc2Vec). In our evaluation, we compare DoSeR to several other state-of-the-art approaches on a wide range of different document structures (e.g., tables) and types (e.g., news documents, tweets), and domains (e.g., general and biomedical domains). We also discuss the influence of the quality of the KB on the EL accuracy and compare our results to those of other non-publicly-available, state-of-the-art algorithms.

**Chapter 9** summarizes the findings and contributions of this work. We also briefly discuss the limitations of our conducted experiments. Finally, we provide an outlook and possible future developments to conclude this work.

## 1.5 Publications

The following full papers, posters and survey articles have been accepted and published in the context of this work. All works listed are referenced again in the respective sections of this work.

***Full Papers:***

- **Do We Need Entity-Centric Knowledge Bases for Entity Disambiguation?**
  In this paper, we provided a systematic evaluation of search-based EL approaches along four variables: (i) the representation of the KB as being either entity-centric or document-centric, (ii) the size of the KB in terms of entities covered, (iii) the semantic heterogeneity of a domain, and (iv) the quality and completeness of a KB. Our results suggest that domain-heterogeneity, size and KB quality have to be carefully considered for the design of EL systems.

  > Zwicklbauer, Stefan et al.: 'Do We Need Entity-Centric Knowledge Bases for Entity Disambiguation?' *13th International Conference on Knowledge Management and Knowledge Technologies, I-KNOW '13, Graz, Austria, September 4-6, 2013.* 2013: 4:1–4:8

- **Search-based Entity Disambiguation with Document-Centric Knowledge Bases**
  In this work, we investigated how the quantity of annotated entities within documents and the document count used for entity classification influence EL results. Our results show that search-based, document-centric EL systems must be carefully adapted with reference to the underlying domain and availability of user data.

  > Zwicklbauer, Stefan et al.: 'Search-based Entity Disambiguation with Document-centric Knowledge Bases'. *Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business, I-KNOW '15, Graz, Austria, October 21-23, 2015.* 2015: 6:1–6:8

- **From General to Specialized Domain: Analyzing Three Crucial Problems of Biomedical Entity Disambiguation**
  In this work, we investigated three crucial properties of specialized EL systems: (i) the entity context (i.e., entity-centric or document-centric KB), (ii) user data, i.e., the quantity and quality of externally disambiguated entities, and (iii) the quantity and heterogeneity of the entities to disambiguate. Our results indicate that EL systems must be carefully adapted when expanding their KBs with special domain entities.

  > Zwicklbauer, Stefan et al.: 'From General to Specialized Domain: Analyzing Three Crucial Problems of Biomedical Entity Disambiguation'. *Database and Expert Systems Applications - 26th International Conference, DEXA 2015, Valencia, Spain, September 1-4, 2015, Proceedings, Part I.* 2015: pp. 76–93

- **DoSeR - A Knowledge-Base-Agnostic Framework for Entity Disambiguation Using Semantic Embeddings**
  In this work, we proposed the DoSeR framework (Disambiguation of Semantic Resources), a publicly available EL framework that is KB-agnostic in terms of RDF-based (e.g., DBpedia) and entity-annotated document KBs (e.g., Wikipedia). DoSeR automatically generates semantic entity embeddings from a set of given KBs first,

and then accepts documents annotated with surface forms and collectively links them to an entity using a graph-based approach.

> Zwicklbauer, Stefan et al.: 'DoSeR - A Knowledge-Base-Agnostic Framework for Entity Disambiguation Using Semantic Embeddings'. *The Semantic Web. Latest Advances and New Domains - 13th International Conference, ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Proceedings*. 2016: pp. 182–198

- **Robust and Collective Entity Disambiguation Through Semantic Embeddings**
  In this work, we presented a new robust and collective, state-of-the-art EL algorithm that uses semantic entity and document embeddings. Our algorithm is also able to abstain if no appropriate entity is available for a given surface form. Our evaluation revealed that our approach (significantly) outperforms other publicly and non-publicly-available EL algorithms on most data sets without data-set-specific tuning.

  > Zwicklbauer, Stefan et al.: 'Robust and Collective Entity Disambiguation Through Semantic Embeddings'. *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '16. Pisa, Italy: ACM, 2016: pp. 425–434

***Posters:***

- **Towards Disambiguating Web Tables**
  In this work, we proposed a methodology to annotate table headers with semantic type information based on the content of columns' cells. We found that in the column header annotation task, for 94% of the maximal F1 score, only 20 cells (37%) need to be considered on average.

  > Zwicklbauer, Stefan et al.: 'Towards Disambiguating Web Tables'. *Proceedings of the ISWC 2013 Posters & Demonstrations Track, Sydney, Australia, October 23, 2013*. 2013: pp. 205–208

***Surveys:***

- **Linking Biomedical Data to the Cloud**
  In this survey, we reviewed state-of-the art EL approaches in the biomedical domain. The main focus lied on annotated corpora (e.g., CalbC), term EL algorithms (e.g., abbreviation disambiguation), and gene and protein EL algorithms (e.g., inter-species gene name disambiguation).

  > Zwicklbauer, Stefan et al.: 'Linking Biomedical Data to the Cloud'. *Smart Health - Open Problems and Future Challenges*. 2015: pp. 209–235

# Part II

Related Work

# CHAPTER 2

## Introduction to Entity Linking

Entity Linking (EL) has been extensively studied in the last decade. During this time different task descriptions have evolved and various challenges came up that contributed to improve the state-of-the-art. In this chapter, we give a brief introduction to EL and its preliminaries. More specifically, we provide an overview of EL including different problem formulations, challenges and related tasks that have evolved in Section 2.1. In Section 2.2, we briefly present typical knowledge bases (KB) that have been used as entity databases for EL. Finally, in Section 2.3, we show how EL algorithms have been evaluated in the literature. The notation introduced in this section is used throughout this work.

### 2.1 Entity Linking - An Overview

Important research areas, such as Information Extraction, Information Retrieval, Machine Translation and Content Analysis, strongly benefit from resolving ambiguities in words. Moreover, different research communities, such as the Natural Language Processing, Semantic Web and Data Mining community, have addressed the problem of word ambiguity, which has been framed in different ways [Hac13]: EL (in the focus of this work), Word Sense Disambiguation (WSD), (Cross-Document) Co-Reference Resolution and Record Linkage. The research communities introduced a plethora of algorithms and approaches, but tackled the four tasks separately so far, often duplicating efforts and solutions [Mor14].

In the following, we provide an EL task description in Section 2.1.1, give a brief overview of EL challenges that contributed to improve the state-of-the-art in Section 2.1.2 and propose other related tasks that found significant attention in the literature in Section 2.1.3.

#### 2.1.1 Entity Linking - Problem Formulation

The task of EL is to establish links between previously identified surface forms (often denoted as entity mentions) and entities within a KB by resolving the probem of *semantic ambiguity* [Zwi15b]. EL inherently involves resolving many-to-many relationships. That is, several surface forms may refer to the same entity (i.e., *synonymy*). Additionally, multiple surface forms may refer to distinct entities (i.e., *polysemy*) [Bag98b]. Figure 2.1 depicts an example of polysemy and synonymy. A sentence contains the surface forms 'Ford' and 'CART'. Both surface forms (underlined) may refer to different entities. For instance, 'Ford' could be an actor (entity *Harrison Ford*), the 38th president of the United States (entity *Gerald Ford*), an organization (entity *Ford Motor Company*) or a place (entity *Ford Island*). In this specific example, we assume *Gerald Ford* to be the correct entity. Simultaneously, this entity can be expressed in several ways, e.g., 'Gerald Rudolph Ford, Jr' or 'G. Ford' [Zwi15b]. Although EL has been well researched so far, there is still confusion

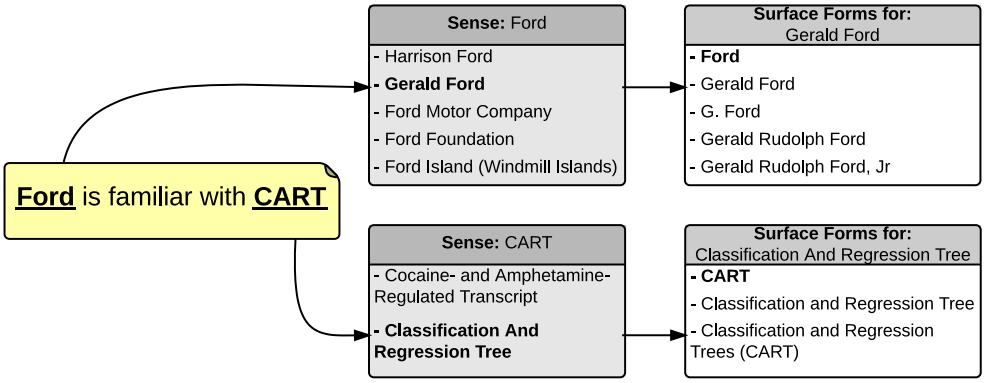



**Figure 2.1:** Surface forms (underlined) within a sentence (yellow rectangle) may refer to different entities (rectangles in the middle) depending on the context. Additionally, an entity may be addressed by various surface forms (rectangles on the right) [Zwi15b].

about the task itself. One reason for that is because there is no standard definition of the problem [Lin15]. For instance, no annotation guidelines are defined: What types of entities or entity classes are valid linking targets? Another confusion occurs when multiple entities are plausible for annotation. How many or which one should be chosen?

Despite these confusions, most authors of EL systems defined the EL task as follows (e.g., [Guo14; Ji10; She15]):

**Definition 2.1. (Entity Linking (EL))** Let $M =< m_1,...,m_S >$ be a tuple of $S$ surface forms in a document $d$, let $C =< c_1,...,c_S >$ be a tuple of the surface forms' textual contexts in $d$ and let $\Omega = \{e_1,...,e_{E-1}, NIL\}$ be a set of $E$ target entities. The task of EL is to find an optimal entity assignment $\Gamma =< t_j^1,...,t_k^S >$ with $t_j^i \in \Omega$, where $t_j^i$ is the assigned entity $e_j$ for surface form $m_i$.

In this definition, EL depends on the preceding Entity Recognition step, during which the boundaries of surface forms are identified. Some researchers [Guo13; Sil13] suggested to incorporate Entity Recognition into the EL task to jointly identify and link entities in documents. It has been shown that this a promising step to improve the annotation accuracy in documents where Entity Recognition tools perform poorly (e.g., twitter tweets).

Some subtasks have evolved over time that are very similar to the EL task. One such task is called Entity Disambiguation [Alh14b]:

**Definition 2.2. (Entity Disambiguation)** Let $M =< m_1,...,m_S >$ be a tuple of $S$ surface forms in a document $d$, let $C =< c_1,...,c_S >$ be a tuple of the surface forms' textual contexts in $d$ and let $\Omega = \{e_1,...,e_E\}$ be a set of $E$ target entities. The task of Entity Disambiguation is to find an optimal entity assignment $\Gamma =< t_j^1,...,t_k^S >$ with $t_j^i \in \Omega$, where $t_j^i$ is the assigned entity $e_j$ for surface form $m_i$.

In contrast to the Entity Disambiguation task, EL algorithms have to cope with the situation that no appropriate candidate entity is in $\Omega$. Entity Disambiguation algorithms assume that the correct target entity is available in the KB.

Another very similar task is Wikification, which exclusively focuses on disambiguating Wikipedia entities [Mih07]:

**Definition 2.3. (Wikification)** Let $\Omega = \{e_1,...,e_W\}$ be a set of $W$ Wikipedia entities (articles). Given an input document $d$, the task of Wikification is to identify all surface forms $M =< m_1,...,m_S >$ in $d$, with $S$ denoting the number of surface forms in $d$. Further, the task is to find an optimal assignment $\Gamma =< t_j^1,...,t_k^S >$ with $t_j^i \in \Omega$, where $t_j^i$ is the assigned Wikipedia entity $e_j$ for surface form $m_i$.

The task of Wikification includes the recognition of phrases that refer to Wikipedia entities (articles). Hence, in this definition, the annotation of the pseudo-entity $NIL$ is not necessary. We note that there is often a lot of confusion about these tasks, because the terms EL, Entity Disambiguation and Wikification are often used interchangeably since the main task of linking surface forms to entities in a KB is the same.

Moreover, in the biomedical domain, the task of EL has often been referred to as Gene Normalization. Gene Normalization is the task of automatically linking surface forms in scientific literature to unique gene or protein identifiers [Mor08b].

Apart from these descendants, many EL approaches narrow the number of relevant entities down by linking **named entities** only due to their frequent occurrences in documents and the massive amount of knowledge about them in the respective KBs. In the general domain, named entities are commonly categorized into the following three entity subgroups: persons, organizations and locations. In the biomedical domain, genes and proteins typically represent named entities. As a consequence, EL and Entity Disambiguation are also known as Named EL and Named Entity Disambiguation, respectively, if the focus lies on exclusively disambiguating named entities.

Additionally, EL can be distinguished between **mono-lingual** and **cross-lingual** EL. Cross-lingual EL refers to linking a surface form in a background source document in one language with the corresponding entity in a KB written in another language [Zha13a]. However, most effort has been expended in mono lingual EL in English language. In this work, we exclusively focus on mono-lingual EL.

In the following subsections, we introduce two important EL challenges that contributed to improve the state-of-the-art (Section 2.1.2) and propose EL related tasks that have found significant attention in the literature (Section 2.1.3).

### 2.1.2 Entity Linking Challenges

In the following, we present two well-known EL challenges in the general and biomedical domain, which significantly contributed to improving state-of-the-art EL systems:

- **TAC-Knowledge Base Population:** A very popular EL challenge is the shared task challenge proposed by the National Institute of Standards and Technology (NIST) as part of the Knowledge Base Population (KBP) track within the Text Analysis Conference (TAC)[1] in 2009 [Alh14b; McN09]. In the KBP-EL task, each EL system obtains a KB, a set of queries consisting of exactly one surface form

1 http://www.nist.gov/tac/, last accessed on 2016-11-28

and the corresponding article in which the surface form appears. The underlying TAC-KBP specific KB comprises a subset of English Wikipedia entities (articles) from an older Wikipedia dump. A participating system has to either disambiguate the given surface form or return the pseudo-entity *NIL* in case of no entity being relevant. The NIST data set is not suited for collective EL since only one surface form is given per input query. In the recent years, the task description has been evolved and expanded. For instance, surface forms have to be extracted from a multi-lingual document corpus and linked to the corresponding entity in a KB (i.e., cross-lingual EL). Most authors, who did not evaluate their EL approach in the context of the TAC conference, used the evaluation data sets proposed in 2009 [McN09], 2010 [Ji10] or 2011 [Ji11b] and compared the results to the best systems of the respective challenge. Unfortunately, the data sets are not open source or freely available. As a summary, Ji and Grishman [Ji11a] published a detailed overview of the TAC-KPB state-of-the-art approaches and its results.

- **BioCreative Gene Normalization Task:** BioCreative[1] (Critical Assessment of Information Extraction for Biology) is a community-wide effort to advance the research on text mining and information extraction systems in the biomedical domain. BioCreative conducted several challenges and released appropriate evaluation data sets for different tasks. To promote the successful development of EL systems in terms of different name variations and ambiguity degrees, BioCreative held several different competitions for the Gene Normalization task [Hir05; Lu11; Mor08b]. This task evaluates the ability to generate gene identifiers from PubMed articles[2]. More specifically, in contrast to the TAC-KBP task, the BioCreative Gene Normalization task demands to recognize all surface forms mentioned in a given article and to (collectively) link them to their corresponding entity identifiers in a KB. All released data sets of the Gene Normalization task are freely available for non-commercial use.

### 2.1.3 Related Tasks

Some other EL related tasks have been well researched, which also tackle the problem of word ambiguity. In the literature, the following three tasks were established besides EL:

- **Word Sense Disambiguation:** WSD identifies the meaning of words (i.e., nouns, verbs and adjectives) in a document [McC03; Nav09]. The main difference between EL and WSD is the kind of inventory used: WSD relies on dictionaries, while EL makes use of entity-defining KBs. Moreover, in EL, in contrast to WSD, a surface form may be partial while still being unambiguous thanks to the context [Mor14]. Regarding Example 2.1, the verb 'play' can be disambiguated by selecting the game/soccer playing sense in a dictionary; on the other hand, the surface form 'Munich' is partial and ambiguous and can be linked to the correct entity in a KB, which would be *FC Bayern Munich*.

1 http://www.biocreative.org/, last accessed on 2016-11-28
2 http://www.ncbi.nlm.nih.gov/pubmed, last accessed on 2016-11-28

**Example 2.1.** Thomas Müller **plays** for **Munich**.

Overall, the EL and WSD tasks are very similar, both involve the disambiguation of textual fragments according to a reference inventory/KB. An in-depth survey of WSD can be found in [Nav09].

- **(Cross-Document) Coreference Resolution:** Anaphora resolution or as it has also been known since the Message Understanding Initiative (MUC), Coreference Resolution or Entity Resolution, is the task of identifying which parts of a text refer to the same discourse entity. The rationale for this task is that the same entity can be referred to in texts through different linguistic expressions [Poe11]. For instance, 'Bush', 'Mr. President', 'G. W. Bush', and 'he' occurring in a document might refer to the same entity [Rao13]. An expansion of the task is called *Cross-document Coreference Resolution*, which describes the identification of the same entity across several documents. Cross-document Coreference Resolution differs from within-document Coreference Resolution in a substantial way. Within a document there is a certain amount of linguistic regularities and consistencies that cannot be expected across documents [Bag98b]. An in-depth survey of Coreference Resolution can be found in [Poe11].
- **Record Linkage:** In the real world, entities usually have two or more representations in databases. Duplicate records do not share a common key and/or they contain errors that make duplicate matching a difficult task [Yak10]. Hence, the goal of Record Linkage (also known as duplicate detection, entity matching and reference reconciliation) is to match records from multiple databases that refer to the same entities, such as matching two publication records referring to the same paper [She15]. Most Record Linkage approaches are based on the assumption that duplicates provide the same context in form of similar attribute values. Thus, they typically leverage all available context information of the records (e.g., [Don05]). Anyway, Elmagarmid et al. [Elm07] reviewed the current state-of-the-art Record Linkage approaches in their survey.

Although EL, WSD, Coreference Resolution and Record Linkage share a lot of similarities, we exclusively focus on EL in this work. Details of WSD and Coreference Resolution can be found in the respective surveys.

## 2.2 Knowledge Bases

A KB is the fundamental component of an EL algorithm. KBs define the entity target set and provide different kinds of entity related information. Generally, an entity can be defined intensionally, i.e., through a description, or extensionally, i.e., through instances and usage [Ogd23]. Intensional definitions can be understood as a thesaurus or logical representation of an entity, as it is provided by graph-based KBs, e.g., Resource Description Framework (RDF) KBs. Extensional definitions resemble information on the usage context of an entity, as it is provided by entity-annotated documents [Zwi15a]. Based on these formulations, we roughly distinguish between **entity-centric KBs** (intensional entity

definitions) and **document-centric KBs** (extensional entity definitions) in this work. More formally, we define an entity-centric KB as follows:

**Definition 2.4. (Entity-centric KB (Ent))** An entity-centric knowledge base $KB_{Ent}$ describes a set of $E$ entities $\Omega = \{e_1,...,e_E\}$, with each entity $e_j$ having a primary key $ID$ and a variable number of fields $k$ containing domain-specific attributes [Zwi15a].

In contrast, we define a document-centric KB as follows:

**Definition 2.5. (Document-centric KB (Doc))** A document-centric knowledge base $KB_{Doc}$ contains a set of $N$ natural language text documents $D = \{d_1,...,d_N\}$ and a set of $E$ entities $\Omega = \{e_1,...,e_E\}$. Each document $d_k \in D$ contains a list of $S$ surface forms $M =< m_1,...,m_S >$, with each surface form $m_i$ being assigned to a target entity $t_j^i$, with $t_j^i \in \Omega$ [Zwi15a].

Apart from these KB definitions, the community distinguished KBs for EL according to the domain of the underlying entities (general-domain vs. special-domain KBs). Figure 2.2 presents a classification of KBs that were (often) used in the context of EL in the literature. The respective general-domain KBs are described in Section 2.2.1 and special-domain KBs are presented in Section 2.2.2.

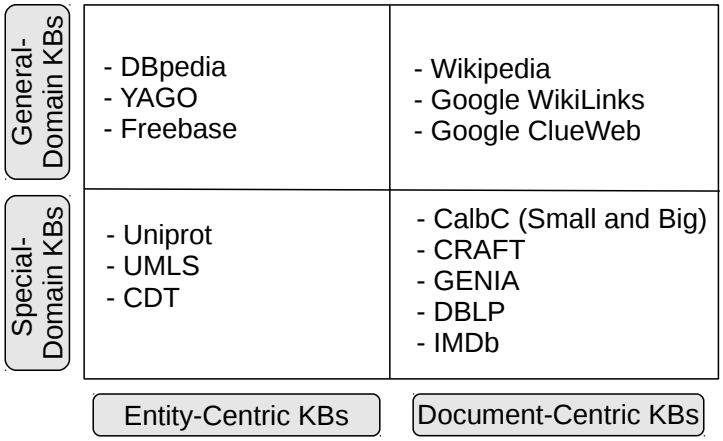



**Figure 2.2:** Classification of popular KBs into general-domain and special-domain KBs as well as entity-centric and document-centric KBs

### 2.2.1 General-Domain Knowledge Bases

Much effort has been expended in linking surface forms to entities located in general-domain KBs. A general-domain KB attempts to cover various entities from the entire world [Des13]. A lot of works additionally narrowed the target entity set down by focusing on named entities only when using general-domain KBs (e.g., [Bar14; Bun06; Hof11]).

However, the following general-domain, **document-centric KBs** have often been used to link entities:

- **Wikipedia**[1] is a document-centric KB and a very popular choice as underlying entity source for EL algorithms. It is a free online, multi-lingual Internet encyclopedia created through decentralized, collective efforts of thousands of volunteers around the world [Bun06]. In its current version, Wikipedia contains more than 5 million article pages, with each article describing a specific concept (entity) in natural language text, tables and figures. Besides, the Wikipedia KB contains a set of valuable features for EL, such as disambiguation pages, redirect pages, an entity category system and interlinks between Wikipedia pages in the entity describing text. In the context of EL, the linkage of surface forms to Wikipedia pages is a crucial step in the Wikification task.
- **Google Wikilinks**[2] is a large entity annotated text corpus comprising 40 million disambiguated surface forms within over 10 million web pages. The surface forms were found by searching textual terms or phrases that closely match with titles of Wikipedia entities. Finally, the surface forms were automatically linked to the respective Wikipedia entities if the annotation system had enough evidences for the correct target entity (i.e., optimized for a high precision).
- **Google ClueWeb**[3] is an extremely large annotated corpus comprising 800 million documents with over 11 billion references to Freebase entities. More specifically, Google annotated the 'ClueWeb09 FACC[4]' and 'ClueWeb12 FACC[5]' corpora with Freebase entities in a fully automatic way, while focusing on optimizing for precision over recall.

The success of Wikipedia and the vision of Linked Open Data has facilitated the automated construction of machine-understanding KBs about the world's entities, their semantic categories and the relationships between them [She13]. Such kind of notable endeavors, i.e., **entity-centric KBs**, which have been extensively used in EL, are:

- **DBpedia** [Aue07] is a multi-lingual, RDF-based KB that contains extracted, structured information from Wikipedia like info boxes, category information, geo-coordinates and external links. The current English language version of the DBpedia KB describes 4.58 million things, out of which 4.22 million are classified in a consistent ontology[6]. Since, DBpedia is a descendant of Wikipedia, it evolves as Wikipedia is getting updated.
- **YAGO** [Suc07] (Yet Another Great Ontology) is a huge semantic RDF-based KB, derived from Wikipedia (e.g., entity categories, redirect pages, info boxes), WordNet [Fel98] and GeoNames[7]. WordNet is a large lexical database where English

1 http://en.wikipedia.org/, last accessed on 2016-11-28
2 http://code.google.com/archive/p/wiki-links/, last accessed on 2016-11-28
3 http://research.googleblog.com/2013/07/11-billion-clues-in-800-million.html, last accessed on 2016-11-28
4 http://lemurproject.org/clueweb09/FACC1/, last accessed on 2016-11-28
5 http://lemurproject.org/clueweb12/FACC1/, last accessed on 2016-11-28
6 http://dbpedia.org/, last accessed on 2016-11-28
7 http://www.geonames.org/, last accessed on 2016-11-28

nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. These synsets are interlinked by means of conceptual-semantic and lexical relations[1]. However, YAGO has knowledge of more than 10 million entities and contains more than 120 million entity facts which makes it particularly suitable for the EL task.

Meanwhile, the authors released YAGO3 [Mah15], an extension of the YAGO KB that combines the information from the Wikipedias in multiple languages. Additionally, it enlarges the original YAGO KB by 1 million new entities and 7 million new facts.

- **Freebase** [Bol08] is a large collaborative, online entity-centric KB consisting of structured metadata created mainly by its community members. Freebase contains more than 43 million entities and 2.4 billion facts about them. The contained data was harvested from sources like Wikipedia and MusicBrainz[2]. Initially, it was developed by the software company Metaweb which was acquired by Google in 2010. In 2014, Google announced that it would shut down Freebase over the succeeding six months and help with the move of the data from Freebase to Wikidata. By now, Freebase data is still available for download[3].

In addition to the presented KBs, other general-domain KBs like ProBase [Wu12] or OpenCyc[4] exist but have not played an important role in EL so far.

### 2.2.2 Special-Domain Knowledge Bases

In contrast to general-domain KBs, special-domain KBs capture concepts, instances and/or relationships of relatively well-defined domains of interest [Des13]. Most notably, the biomedical domain has received much attention in the area of linking proteins and genes (in the biomedical text processing community, the task is more commonly known as normalization). The following **document-centric KBs** are particularly prominent in terms of domain-specific EL:

- **CalbC**[5] (Collaborative Annotation of a Large Biomedical Corpus) is a biomedical, document-centric KB, representing a very large, community-wide shared text corpus annotated with biomedical entity references [Kaf12; Reb10]. CalbC represents a silver standard corpus which results from the harmonization of automatically generated annotations and is freely accessible [Zwi15b]. The data set comprises two differently sized main corpora: CalbCSmall (174 999 documents) and CalbCBig (714 282 documents). All documents located in the CalbC corpora are Medline abstracts of the 'Immunology' domain, a subdomain within the biomedical domain. The documents in CalbCBig contain ≈ 10 million annotated surface forms referring to 228 744 unique entities. These entities can be categorized into four main classes: Proteins and genes, chemicals, diseases and disorders as well as living beings.

1 http://wordnet.princeton.edu/, last accessed on 2016-11-28
2 http://musicbrainz.org/, last accessed on 2016-11-28
3 http://www.freebase.com, last accessed on 2016-11-28
4 http://www.cyc.com/platform/opencyc/, last accessed on 2016-11-28
5 http://www.calbc.eu/, last accessed on 2016-11-28

- **GENIA** [Kim03] contains ≈ 2000 MEDLINE abstracts from the domain of molecular biology and was released in 2003. All MEDLINE abstracts were collected by querying PubMed[1] for the three MeSH terms 'human', 'blood cells', and 'transcription factors'. The abstracts were syntactically and semantically annotated, resulting in six different sub-corpora corresponding to the specific annotations. One of these sub corpora was annotated with ≈ 89 000 entity references to the GENIA ontology [Zwi15b].
- **CRAFT** [Bad12] (Colorado Richly Annotated Full Text) comprises 67 full-text journal articles from the biomedical domain. Overall, the corpus contains ≈ 100 000 annotations from the biomedical domain, linking it to seven different repositories (Chemical Entities of Biological Interest, Cell Ontology, Entrez Gene, Gene Ontology, NCBI Taxonomy, Protein Ontology and Sequence Ontology) [Zwi15b].

With EL in the biomedical domain being a well-researched topic, other, less extensive and even more specialized KBs have been used. However, a detailed overview of other document-centric KBs in the biomedical domain can be found in [Zwi15b]. Works that focused on other domains, such as computer science or movies, exist but are much less used in the context of EL. In the following, we present two well-known, domain-specific, document-centric KBs that cover these two domains:

- **DBLP**[2] (Digital Bibliography & Library Project) is an online computer science bibliography containing journal articles, conference papers, and other publications. Overall, the KB contains five types of objects (entities): papers, authors, publication venues, title terms and publications years. The DBLP network contains over 1.24 million authors, 2.6 million publications and 7000 venues (conferences/journals).
- **IMDb**[3] (Internet Movie Database) is a movie database launched in 1990 and is currently a subsidiary of Amazon[4]. It contains information related to films, television programs including cast, production crew, fictional characters, biographies, summaries and reviews. As of February 2016, the KB comprised approximately 3.6 million titles (including episodes) and 7 million personalities.

Overall, manually curated, document-centric KBs are often rare since a huge effort is necessary to provide a high number of high-quality annotations in documents.

In contrast, several **entity-centric KBs** have evolved to describe entities in specialized domains:

- **UMLS**[5] (Unified Medical Language System) is a compendium of many vocabularies and classifications in the biomedical domain. UMLS comprises the following three components: the Metathesaurus, the Semantic Network, and the SPECIALIST

1 http://www.ncbi.nlm.nih.gov/pubmed, last accessed on 2016-11-28
2 http://www.dblp.org, last accessed on 2016-11-28
3 http://www.imdb.com, last accessed on 2016-11-28
4 http://www.amazon.com, last accessed on 2016-11-28
5 http://www.nlm.nih.gov/research/umls/, last accessed on 2016-11-28

Lexicon. The Metathesaurus forms the base of UMLS, which contains over 1 million biomedical concepts and 5 million concept names. Each concept has its specific attributes that define the entity's meaning, provides relations to other related entities and is linked to corresponding entities of other source vocabularies.

- **UniProt** [Mag11] is a comprehensive KB providing high-quality resources of protein sequences. The core KB (UniProtKB) is the central hub for the collection of functional information on proteins, with accurate, consistent and rich annotations[1]. UniProtKB consists of two subsections: Swiss-Prot and TrEMBL. Swiss-Prot is a manually annotated protein sequence database curated with information extracted from scientific literature and comprises about 550 000 entity entries. In contrast, TrEMBL contains about 61 million computationally and automatically analyzed entries. The UniProt KB is particularly relevant for biomedical EL, since a bulk of works are specialized on exclusively linking genes and proteins. This constitutes a challenging task due to a high degree of ambiguous gene/protein mentions across species [Che05].
- **CDT** [Dav16] (Comparative Toxicogenomic Database) is a publicly available, entity-centric KB[2] that describes 14 672 chemicals, 6401 diseases and 42 761 genes. In particular, it contains three types of manually curated facts: 202 085 chemical-disease associations, 33 583 gene-disease associations and 1 379 105 chemical-gene interactions.

Several other domain-specific KBs exist that provide less entities or facts. These KBs were hardly used in the context of EL due to the limited number of entities and the necessity to strongly adapt EL methods to fully exploit the underlying entity information of the KB. It is still an open problem how to leverage the different types of knowledge from different KBs without explicitly adapting the underlying algorithms.

## 2.3 Evaluation of Entity Linking Systems

The evaluation of EL systems is a crucial factor, in particular when we compare different systems. An evaluation demands one or multiple data sets that are enriched with ground truth annotations (i.e., allegedly correct entity assignments that are used for comparison with the output of an EL system). Throughout this work, we use variable $e_j^{m_i} \in \Omega$ to denote that entity $e_j$ is the ground truth annotation of surface form $m_i$. Most EL works make use of the well-known standard measures recall, precision, F1 and accuracy. In the literature, the evaluation measures have been defined slightly different in the context of EL, depending on how NIL annotations are evaluated.

In this work, we use the definitions proposed by Cornolti et al. [Cor13] and Usbeck et al. [Usb15]. In the following, we let function $f(a, b)$ return whether the entities $a, b \in \Omega$

1 http://www.uniprot.org/, last accessed on 2016-11-28
2 http://ctdbase.org/, last accessed on 2016-11-28

are identical and let function $nnil(a)$ return whether entity $a \in \Omega$ is not $NIL$:

$$f(a,b) = \begin{cases} 1, & \text{if } a = b. \\ 0, & \text{otherwise.} \end{cases} \qquad nnil(a) = \begin{cases} 1, & \text{if } a \neq NIL. \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

Given a document's ground truth assignment $G =< e_j^{m_1},...,e_k^{m_S} >$ with $e_j^{m_i} \in \Omega$ and $S$ denoting the number of surface forms in this document, and given an entity assignment $\Gamma =< t_j^1,...,t_k^S >$ with $t_j^i \in \Omega$, then recall, precision and F1 are defined as follows:

$$Recall = \frac{\sum_i^S (nnil(e_k^{m_i}) \wedge f(e_k^{m_i}, t_j^i))}{\sum_i^S nnil(e_k^{m_i})} \tag{2.2}$$

$$Precision = \frac{\sum_i^S (nnil(t_j^i) \wedge f(t_j^i, e_k^{m_i}))}{\sum_i^S nnil(t_j^i)} \tag{2.3}$$

$$F1 = \frac{2 * Recall * Precision}{Recall + Precision} \tag{2.4}$$

At the moment, these measures ignore NIL annotations. For instance, if a data set contains a NIL ground truth annotation, it will not be considered during the recall computation, i.e., the respective output of the underlying EL system will not be evaluated. Moreover, if an EL system returns a NIL annotation, it will not be evaluated during the precision computation, i.e., the respective ground truth annotation will be ignored. To explicitly evaluate an EL system in terms of entity and NIL annotations, we compute the accuracy as follows:

$$Accuracy = \frac{\sum_i^S f(t_j^i, e_k^{m_i})}{S} \tag{2.5}$$

Throughout this work, we use the definitions given above if the usage of other measures is not emphasized. Most authors published micro-F1 values (and micro-recall and micro-precision values respectively), where each surface form is considered separately and is seen as equally important. However, few authors published macro-F1 values and simply called them F1-measures. Macro-F1 values are computed across the set of documents instead of surface forms, which might lead to (significant) result discrepancies [Man08]. Thus, it is crucial to distinguish between both evaluation metrics.

The authors of the works [Che13; Mil08b; Rat11] employed the evaluation methodology BOT (bag-of-titles) to evaluate Wikification tasks. Here, a system has to identify all surface forms in a document and link them to the appropriate Wikipedia article. When using BOT, the outputs of an annotation system are compared to the ground truth annotations for that document while ignoring duplicate candidates. The evaluation measures are recall, precision

and F1. In the BOT evaluation, the set of titles (i.e., entities) annotated in the ground truth are collected. Taking the example of Ratinov et al. [Rat11], we assume that the ground truth annotations are {('China', *People's Republic of China*), ('Taiwan', *Taiwan*), ('Jiangsu', *Jiangsu*)}. Further, we assume that the predicted annotations are {('China', *People's Republic of China*), ('China', *History of China*), ('Taiwan', *null*), ('Jiangsu', *Jiangsu*), ('republic', *Government*)}. Given the ground truth annotations, the BOT is {*People's Republic of China*, *Taiwan*, *Jiangsu*} and the BOT for the predicted annotation is: {*People's Republic of China*, *History of China*, *Jiangsu*}. The entity *Government* is not included in the BOT for the predicted annotations because its associated surface form 'republic' does not appear as a surface form in the ground truth annotations [Rat11].

Another evaluation metric has been used in the TAC-KBP tasks since 2011. Basically, an EL system has to cluster the queries, i.e., surface forms that refer to the same entity, and decide whether a cluster refers to an entity in the KB [Ji11b]. To this end, Ji and Grishman proposed a modified *B-Cubed* [Bag98a] metric called *B-Cubed+* to evaluate these clusters. For a detailed explanation, we refer the interested reader to the original work [Ji11b].

However, a vast number of research works in EL, Entity Disambiguation and Wikification led to non-uniform terminology and non-comparable evaluation metrics and techniques. To overcome this deficit, Cornolti et al. [Cor13] implemented a publicly available benchmarking framework for EL systems that provides an overview of the efficiency and effectiveness of EL algorithms[1]. Providing different data sets and their ground truth annotations in the background, EL systems are queried with data set documents which have to be annotated and returned to the benchmarking system. Usbeck et al. [Usb15] proposed the publicly available EL benchmarking framework *GERBIL*[2], which can be seen as a further development of the framework by Cornolti et al. [Cor13]. More specifically, it provides persistent URLs for experimental settings, which also addresses archiving experimental results. To tackle the problem of reproducibility, the results of GERBIL are published in a machine-readable format. GERBIL is the current state-of-the-art benchmarking framework to evaluate Information Extraction tasks like EL, (Named) Entity Recognition and EL, or Entity Typing. Anyway, in this work, we mostly utilize the GERBIL framework to evaluate our approaches.

1 http://acube.di.unipi.it/, last accessed on 2016-11-28
2 http://aksw.org/Projects/GERBIL.html, last accessed on 2016-11-28

# CHAPTER 3

## Entity Linking Approaches

Installed and ready-for-use Entity Linking (EL) systems accept documents with one or multiple labeled surface forms. These surface forms are typically manually annotated or recognized by an Entity Recognition algorithm, which detects the boundaries of potential entities in a document in a preceding step. The Entity Recognition step is often outsourced and not included in EL systems. In this work, we exclusively focus on EL approaches and refer to the survey [Nad07] as well as some recent methods [Fin05; Pas14; Rat09] for further Entity Recognition information. Well-known, publicly available open source solutions for Entity Recognition are Stanford NER[1], OpenNLP[2] and LingPipe[3].

In this chapter, we provide an in-depth overview of existing EL approaches, which can be subdivided into the following three crucial components: (i) Candidate entity generation, (ii) disambiguation (including EL features), and (iii) abstaining. Figure 3.1 depicts an overview and shows the components that are further analyzed in this chapter.

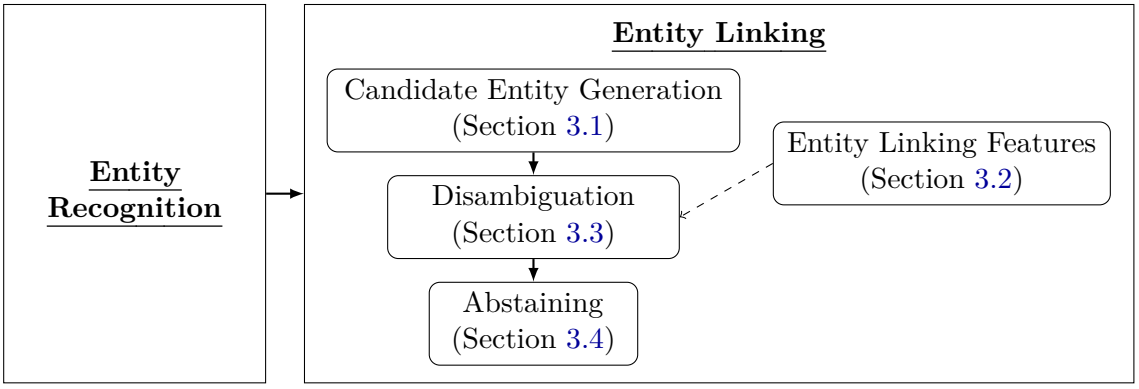



**Figure 3.1:** Typical components of EL approaches

The candidate entity generation step generates a set of relevant entities for each surface form to reduce the number of overall candidates and to optimize the computation process later on. Methods for candidate entity generation are proposed in Section 3.1. After that, in the disambiguation step, disambiguation algorithms establish links between surface forms and relevant target entities by identifying the correct semantic meaning. We first provide

1 http://nlp.stanford.edu/ner/, last accessed on 2016-11-29
2 http://opennlp.apache.org/, last accessed on 2016-11-29
3 http://alias-i.com/lingpipe/, last accessed on 2016-11-29

an in-depth overview of features that are typically used in entity disambiguation algorithms in Section 3.2. Then, we present a rich set of methods to disambiguate candidate entities in Section 3.3. Finally, in Section 3.4, we present abstaining methods, which determine whether the assigned entity is appropriate for the respective surface form. We emphasize that abstaining methods, depending on the approach, are often tied to and integrated in the underlying disambiguation algorithm.

We focus on mono-lingual EL approaches since cross-lingual approaches usually apply the same methods but additionally make use of a dictionary to translate the input documents. However, this is out of scope in this work.

## 3.1 Candidate Entity Generation

At the beginning, the possible target entity set $\Omega_i$ for a surface form $m_i \in M$ in an input document $d$ comprises all entities located in the overall target entity set $\Omega$. The main goal of the candidate entity generation step is to significantly reduce the target entity set $\Omega_i$ for all surface forms $m_i$ to optimize linking accuracy and to reduce computational complexity. As a result, each surface form should exhibit a small set of candidate entities that might be referred by the respective surface form. According to the experiments conducted in [Hac13], candidate entity generation is a fundamental and crucial step for accurate and performant EL systems. However, the step is optional and not integrated in all EL approaches. The remainder of this section provides an overview of state-of-the-art candidate entity generation approaches proposed in the literature. These are name dictionary methods (Section 3.1.1), surface form expansion methods (Section 3.1.2) and search engine methods (Section 3.1.3).

### 3.1.1 Name Dictionary Methods

A name dictionary is the primarily used technique to generate candidate entities. EL approaches build a dictionary that can be seen as a $< key, value >$ structure. The *key* value stores a surface form and the *value* represents a list containing all entities that may be addressed with the surface form *key*. The following Table 3.1 shows an extract of a name dictionary.

**Table 3.1:** Part of a name dictionary

| **Key (Surface Form)** | **Value (Entity)** |
|---|---|
| Apple Inc. | Apple Inc. |
| Jordan | Michael Jordan |
| | Jordan, New York |
| | Jordan River |
| M. Jordan | Michael Jordan |
| | Michael I. Jordan |
| | Michael Jordan (Football) |
| President G. Washington | Georg Washington |

The dictionary is constructed by fully leveraging the features offered by the respective knowledge base (KB) and/or other external resources. For instance, many works that link Wikipedia entities often extract possible surface forms for entities from the following Wikipedia pages: entity pages, redirect pages, disambiguation pages and bold phrases from the first article paragraph [She15]. These kind of features are used by nearly all Wikification systems (e.g., [Bun06; Gat13; Guo13]). However, the most important feature is the information about the entities' usage context. More specifically, manually or automatically entity-annotated documents provide a rich source for relevant surface forms. For instance, Wikipedia articles contain hyperlinks that link to other Wikipedia entities. The anchor text of a link represents a surface form of the target entity and provides a useful source for synonyms and other name variations. In Example 3.1, the surface forms 'TS' and 'New York' refer to the entities *New York Time Square* and *New York City*.

**Example 3.1.** The **TS** has been a **New York** attraction for over a century.

There also exist some external corpora that are annotated with Wikipedia entities. A popular example is the Google Wikilinks corpus[1] providing ≈ 42 million surface forms and ≈ 3 million distinct Wikipedia entity annotations. Further corpora were proposed in [Art10] and [Day08] with both providing ≈ 55 000 annotated Wikipedia entities. Generally, corpora with a bulk of manually annotated entities are rare since a significant human effort is necessary. If entity annotations were automatically created, one has to regard the accuracy of the annotation system.

Given a (name) dictionary, candidate entities are usually determined by exactly matching the query surface forms with those located in the dictionary, while ignoring large and lower case letters. Depending on the domain, capital letters of surface forms may play a crucial role since capital letters can further specify the underlying entity (e.g., gene entities).

One of the major obstacles that make exact term matching insufficient is the problem of term variations. As a consequence, beside exact matching, partial term matching is essential to provide a high recall in candidate entity generation. Tsuruoka et al. [Tsu07] described the following, most frequent term variations:

- Spelling mistakes
- Orthographic variation (e.g., gene 'IL2' and 'IL-2')
- Morphological variation (e.g., 'Transcriptional factor' and 'Transcription factor')
- Roman-Arabic (e.g., 'Leopold 3' and 'Leopold III')
- Acronym-definition (e.g., 'NATO' and 'North Atlantic Treaty Organization')
- Extra words (e.g., 'United States' and 'United States of America')
- Different word ordering (e.g., 'Serotonin receptor 1D' and 'Serotonin 1D receptor')
- Parenthetical material (e.g., 'The Noise Conspiracy' and 'The (International) Noise Conspiracy')

1 http://code.google.com/archive/p/wiki-links/, last accessed on 2016-11-29

These term variants often result from a combination of these and can be very complex. One way to alleviate the problem is to normalize surface forms first [Fan06; Usb14], if no appropriate candidate entities could be found. This includes converting capital letters to lower case, and deleting hyphens and spaces can resolve some of the mismatches caused by orthographic variation [Tsu07]. Some approaches additionally apply a spell checker in the case of misspelled surface forms. For instance, Chen et al. [Che10] applied the Apache Lucene[1] spell checker to obtain the correct surface form. In contrast, Zhang et al. [Zha10a] made use of the Wikipedia built-in feature "Did you mean?", which provides an entity suggestion for a misspelled string (surface form). Several other works exist that correct spelling mistakes by using the spelling correction service supplied by the Google search engine (e.g., [She12b; Zhe10]).

Further, plenty of works apply string similarity measures, such as Levenshtein distance, Hamming distance, Dice score or Skip Bigram Dice score, to match surface forms in documents to surface forms in the dictionary. The application of such approximate string matching methods solves some of the term variation issues listed before. A survey of string matching methods can be found in [Had11].

Other approaches apply more advanced techniques. For instance, Moreau et al. [Mor08a] proposed a robust, generic model based on Soft TF-IDF [Coh03] to show that similarity measures may be combined in numerous ways. Their model outperforms all other evaluated measures on two corpora. However, in the biomedical domain, string similarity measures have been researched extremely well since character changes might lead to different entity interpretations. In this context, Tsuruoka et al. [Tsu07] proposed a logistic-regression-based approach that learns a string similarity measure from a dictionary. The results indicate that the learned measure outperforms all others like Hidden Markov model [Smi03], Soft TF-IDF, Jaro-Winkler distance [Win90] and Levenshtein distance in dictionary look-up tasks. Another work from Rudniy et al. [Rud14] introduced the Longest Approximately Common Prefix (LACP) method for biomedical string comparison. LACP runs in linear time and outperforms nine other string similarity measures like cosine similarity with TF-IDF weights [Sal88], Jaro-Winkler distance [Win90] or Needleman-Wunsch algorithm [Nee70] in terms of precision and performance.

In summary, the most common rules for partial name dictionary matching applied in EL systems include [She15]:

- The entity name is contained in or contains the surface form.
- The entity name exactly matches the first letters of all words in the surface form or vice versa.
- The entity name shares one or more common words with the surface form.
- The entity name is very similar but does not exactly match the surface form.

If a surface form matches a *key* in the dictionary during partial matching by satisfying at least one of the presented rules, all entities that are stored with *key* are added as candidates

1 http://lucene.apache.org/, last accessed on 2016-11-29

entities. A drawback of partial matching might be increased recall values at the cost of (significantly) decreased precision values. Generally, the order in which exact and partial matching methods are applied depends on the respective approach. Typically, a partial matching approach is applied if an exact matching method does not retrieve any candidate entities. Anyway, name dictionary methods for candidate entity generation are used by most EL systems but strongly depend on the quantity and quality of underlying entity data.

### 3.1.2 Surface Form Expansion Methods

Surface forms are often acronyms or parts of their full names. For that reason, some EL approaches apply a surface form expansion technique to determine the original full name variation. After surface extension, these EL approaches typically make use of a named dictionary to generate candidate entities. A simple but effective approach for acronym expansion is to search the surface forms' surrounding textual context with the help of heuristic pattern matching (e.g., [Che10; Leh10]). Typical search pattern are acronyms in parenthesis near the expansion (e.g., 'North Atlantic Treaty Organization (NATO)') or expansions that are in parenthesis adjacent to the acronym (e.g., 'NATO (North Atlantic Treaty Organization)'). Other context-based approaches were proposed by Zheng et al. [Zhe10] and Zhang et al. [Zha10b]. The authors suggested to use the entire document to identify the expanded form with a n-gram-based approach. After stop word removal, they suggested to search for $n$ successive words that start with the same initials as the characters of the acronym. If existent, the $n$ matching words nearest to the acronym are considered as surface form expansion. In contrast, Cucerzan [Cuc11] applied an acronym detector [Jai07], which utilizes information gathered from the web to map acronyms to their full names.

Another approach is to apply an external Entity Recognizer to identify other (named) entities within the document (e.g., [Got11; Var10]). If the surface form of a recognized entity contains the initial surface form as substring, the algorithm considers this entity as an expanded form. For instance, an Entity Recognizer identifies the entity with the surface form 'Michael J. Fox' at the beginning of the input document, then the latter appearing surface form 'Fox' is expanded to the respective full name. If the input document provides multiple surface forms to be linked, expanded forms are occasionally found in other surface forms.

Zhang et al. [Zha11a] introduced a supervised learning method to identify more complicated acronyms (e.g., 'CCP' for 'Communist Party of China'). First, the approach extracts candidate expansions from the document with the help of predefined rules like text markers (such as 'United States (US)' and 'US (United States)') and first letter matching (i.e., word sequences that begin with the same first letter as the acronym and do not contain more than two stop words are identified as candidates). Considering the following Example 3.2,

**Example 3.2.** The Communist Party of China is the founding and ruling political...

the approach extracts 'Communist Party of China is the' for the acronym 'CCP'. The respective substring begins with the letters of the acronym and the phrase ends with two

stop words signalizing to cut off. After extraction, each pair of acronym and possible corresponding expansion is represented as feature vector including part of speech features and the alignment information between the acronym and the expansion [She15]. Finally, a Support Vector Machine (SVM) is applied to score each combination while the highest score candidate is selected as acronym expansion. This approach achieves a statistical significant improvement over state-of-the-art acronym expansion methods.

### 3.1.3 Search Engine Methods

Some works rely on information supplied by search engines such as Google. For instance, Han and Zhao [Han09] queried the Google API with the underlying surface form and its short surrounding context. The results were filtered by domain and all retrieved Wikipedia pages were considered as candidate entities. Dredze et al. [Dre10] chose a very similar approach, but submitted the surface form only and limited the result search space for Wikipedia pages to the top-20 Google results. Lehmann et al. [Leh10] and Monahan et al. [Mon11] stated that the Google search engine is very effective at identifying some of the very difficult mappings between surface forms and entities. Thus, they also made use of the Google API to query the three most relevant candidate entities. Moreover, the authors used the Dice score and acronym tests to ensure that generated Wikipedia candidates are sufficiently similar to the surface form.

Methods based on search engines are much less popular than name dictionary methods due to their significant indexing drawback. When linking Wikipedia or DBpedia entities, search engines ordinarily retrieve the respective entity URLs. However, if more specialized KBs are used whose entities are not available as web pages or are not properly indexed by the search engine, candidate entity generation might be highly incorrect or even not possible.

## 3.2 Entity Linking Features

In this section, we analyze the most relevant features found to be useful and important in terms of ranking candidate entities accurately. The main algorithms that integrate the features to create a full EL system are presented in Section 3.3. In the following, we distinguish between context-independent and context-dependent entity features.

Context-independent features leverage information only from the surface form and candidate entities. These features tend to be very useful and, depending on the domain, might lead to satisfying EL results. However, additional contextual information features are necessary to further improve EL accuracy. In the following, we classify context-independent features into entity name, entity popularity and entity type features, which are utilized by most works.

In contrast, context-dependent features can be divided into textual context features (i.e., features that analyze the surrounding context of surface forms) and topical coherence features (i.e., measuring the relatedness of candidate entities across multiple surface forms to select the most coherent entity assignment). Context-dependent features play a crucial role in all kinds of EL tasks and, hence, a bulk of works were presented that focus on leveraging these features as efficiently as possible. Table 3.2 provides a brief summary of our feature classification and section overview.

**Table 3.2:** Classification of EL features that are discussed in this section

| **Context-independent Features** | **Context-dependent Features** |
|---|---|
| Entity Name (Section 3.2.1) | Textual Context (Section 3.2.4) |
| Entity Popularity (Section 3.2.2) | Topical Coherence (Section 3.2.5) |
| Entity Type (Section 3.2.3) | |
| Joint Feature Modeling (Section 3.2.6) | |

Another way to distinguish between EL features is to define local and global features [Rat11]. Local features typically analyze how good a surface form and its surrounding (textual) context matches a candidate entity (i.e., context-independent features and textual context features). In contrast, global features typically capture the topical coherence in the entire document defined by multiple assigned candidate entities across different surface forms (i.e., topical coherence features). Global features often measure the relatedness between a pair of assigned candidate entities (as proposed in Section 3.2.5).

However, in Section 3.2.6 we discuss techniques that cannot be separated from each other and, thus, jointly model multiple features. For instance, very popular models are topic model.

### 3.2.1 Entity Name

A comparison of surface forms and entity names is the most intuitive and direct feature for candidate entity ranking. For this purpose, various standard string similarity measures, similar to those presented in Section 3.1.1, are used by most approaches for name comparison. Those include cosine similarity with TF-IDF weights [Sal88], edit distances [Liu13; Zhe10], Dice coefficient score [Leh10; Mon11], and left and right Hamming distance scores [Dre10]. Basically, the most common name comparison features include [She15]:

- Whether the surface form exactly matches the candidate entity name.
- Whether the candidate entity name is a prefix or suffix of the surface form, or vice-versa.
- Whether the candidate entity name is an infix of the surface form, or vice-versa.
- Whether all of the letters of the surface form are found in the same order in the candidate entity name.
- The number of same words between the surface form and the candidate entity name.
- The ratio of the recursively longest common subsequence [Chr06] to the shorter among the surface form and the candidate entity name.

Overall, entity name similarity measures are very similar to those proposed in the name dictionary Section 3.1.1. The main difference in these models is the final distance score. While in Section 3.1.1 a binary decision of the form “match” or “no match” is enough, entity name similarity measures return a comparable score to rank the candidate entities appropriately. A survey of basic string similarity measures can be found in [Coh03].

### 3.2.2 Entity Popularity

The entity popularity is another crucial entity describing feature, which is frequently used in EL systems. It is based on the assumption that some entities (e.g., *Influenza*) occur more often than others (e.g., *IIV3-011L gene*). Hence, popular entities tend to re-occur in other documents with a higher probability. Typically, the entity popularity is described as a-priori probability that an entity occurs [Res95]. Generally, the entity popularity (also known as **Entity Prior**) represents the entity occurrence probability $p(e_j)$, which can be computed as follows:

$$p(e_j) = \frac{score(e_j)}{\sum_{e_k \in \Omega} score(e_k)} \tag{3.1}$$

The probability $p(e_j)$ is computed with respect to the abstract function $score(e_j)$, which returns a score for each candidate entity $e_j$. For instance, in the works of Ratinov et al. [Rat11] and Han et al. [Han11a], the *score* function returns the number of hyperlinks within Wikipedia pages to entity $e_j$. In contrast, in Guo et al. [Guo13] and Gattani et al. [Gat13], the *score* function retrieves how often Wikipedia user have visited the respective entities. In contrast, Dredze et al. [Dre10] exploited the entities' Wikipedia graph structure, like the indegree of the node, the outdegree of the node, and the Wikipedia page length in bytes to compute an entity score. Shen et al. [She14] proposed a generic approach to determine the Entity Prior within an arbitrary entity-centric KB. The authors leveraged the link structure of the entity-centric KB by applying the PageRank algorithm [Bri98] and used the nodes' (entities) PageRank score as entity score.

A modification of the prior definition above is the additional consideration of the input surface form. For instance, with respect to the surface form 'Wall Street', the candidate entity *Wall Street (Film)* is much rarer than the candidate entity *Wall Street*. Most time when people mention 'Wall Street', they mean the street in New York City rather than the film. More formally, the conditional probability $p(e_j|m_i)$, also known as **Sense Prior**, defines the popularity feature for a (candidate) entity $e_j$ with respect to the surface form $m_i$. It is defined as the proportion of links with the surface form $m_i$ as the anchor text, which link to entity $e_j$:

$$p(e_j|m_i) = \frac{count_{m_i}(e_j)}{\sum_{e_k \in \Omega} count_{m_i}(e_k)} \tag{3.2}$$

Here, the function $count_{m_i}(e_j)$ returns how often entity $e_j$ has been associated with the given surface form $m_i$.

The priors' quality strongly depends on the quantity of the underlying knowledge source. The more training data is used to compute the priors the more accurate are they reflecting the true entity occurrence probability. Hence, most state-of-the-art EL systems leverage the Wikipedia link information due to its extensive number of $\approx$ 90 million interlinks (e.g., [Hof11; Liu13; She12b]). Other works use the Sense Prior as EL baseline (i.e., link surface forms to the entities with the highest probability), as it performs exceptionally well on many data sets (e.g., [Chi15; Guo14; Rat11]). Ji and Grishman [Ji11a] analyzed that

simple candidate ranking features like the Entity Prior and in particular the Sense Prior can achieve a high EL accuracy. More specifically, their entity popularity ranking method based on Web popularity achieved an accuracy of 0.71%, which outperforms all systems of the TAC-KBP2010 [Ji10] challenge.

Thus, regardless of the domain, the Entity Prior and Sense Prior are important features for effective EL, but completely rely on the number of available entity annotations. For instance, in the specialized gene domain, no work has been found that applies the Sense Prior since the number of entity annotated documents is very limited.

### 3.2.3 Entity Type

In some works, the entity type (e.g., person, organization, location, gene, protein) is used as additional ranking feature to facilitate candidate entity ranking for a given surface form. Here, the surface form type should be consistent with the type of the candidate entity in a KB. Depending on how surface forms are determined, the surface form may be already a-priori restricted to a specific type. For instance, if documents where enriched with annotations by a Named Entity Recognition system, the surface forms typically refer to persons, organizations or locations.

If the surface forms' types remain completely unknown at the beginning of the linking step, the type can be inferred by applying a Named Entity Recognition algorithm as done by Nemeskey et al. [Nem10]. The authors used their in-house approach to identify the entity type of surface forms as well as the types of those entities whose type is unavailable in the KB. Further, Lehmann et al. [Leh10] and Monahan et al. [Mon11] made use of the LCC's CiceroLite Named Entity Recognition algorithm [Var07] to determine the surface forms' types (i.e., persons, organizations and locations). If the candidate entity type is unknown in their underlying KB, the authors consulted DBpedia and LCC's WRATS ontology resources. Another approach to infer the surface form type was presented in [Dre10], where the authors identified the type by matching the surrounding context with the Wikipedia infobox content. Finally, all authors exclusively assigned entities with consistent surface form types.

In the gene domain, the species is an important gene and protein type that can be considered as additional valuable information for EL. For instance, the following example mentions the gene name 'P54' with a variety of genes being considered as potential candidates for this name [Hak08].

> **Example 3.3.** The **P54** gene was previously isolated from the chromosome translocation breakpoint region on 11q23 of **RC-K8** cells...

The cell term 'RC–K8' helps to narrow down the candidate set to the particular 'human' species with several human gene candidate entities still being relevant. Anyway, the number of approaches that resolve species of surface forms ranges from simple rule-based approaches that search for species identifiers in the surrounding textual context [Hak08; Hsi14] to more complex machine learning approaches (e.g., [Wan09; Wan10]). However, since species disambiguation is out of scope in this work we refer to our survey about biomedical entity linking [Zwi15b].

### 3.2.4 Textual Context

The most straightforward textual context feature is to measure the similarity between the textual context $c_i$ around surface form $m_i$ and a representation of a candidate entity $e_j$ [She15] (e.g., entity description). In the context of EL, the following two forms have often been utilized to represent the surrounding contexts of surface forms and entity descriptions:

- **Bag-of-Words:** When using a bag-of-words model, the entire document that contains the surface form (e.g., [Che10; Guo13; Liu13]) or a predefined context window around the surface form (e.g., [Bun06; Han11b; Rat11]) is represented as a bag-of-words. Depending on the type of the underlying KB, the candidate entities' information is also represented as a bag-of-words. This might be the entire entity description (e.g., [Bun06; Liu13; Rat11]), a suitable context window around an occurrence of that entity in a document-centric KB (e.g., [She13]) or from the top-$k$ tokens from an entity summary (e.g., [Che06; Guo13]).
- **Concept Vector:** A lot of works extract various kind of information from the surface form containing document and an entity describing document in a KB to compose a concept vector. These information include previously extracted key phrases (e.g., [Hof12; Ryb14]), named entities (e.g., [Che11; Zha10a]) and descriptive tags (e.g., [Gat13]). Moreover, the description of a candidate entity can be defined according to the data available in the KB. In terms of Wikipedia, this might be other linked entities (i.e., target entities of hyperlinks) in the entity's Wikipedia article and/or relevant facts known through the respective Wikipedia infoboxes (e.g., [Che10; Dre10]).

Based on the formulations above, the respective context features of surface forms and the respective features of candidate entities are often converted to feature vectors. Then, to compute a similarity score between surface form context and candidate entities, various methods have been applied, including cosine similarity (e.g., [Bun06; Che11; She13]), dot product (e.g., [Guo13; Han11b]), word overlap (e.g., [Liu13]), Kullback–Leibler divergence (e.g., [Hof11]), n-gram-based measures (e.g., [Hof11]) and Jaccard similarity (e.g., [Kul09]). However, apart from these mundane features to compute a context similarity, more complex and sophisticated techniques have been proposed.

In the following, we distinguish between language models, topic models and neural network models, and present popular representatives.

#### *Language Models*

A popular language model for EL is the *query language model* [Man08]. In the first step, we infer a language model $M_{e_j}$ for each entity $e_j \in \Omega$. Second, we infer $p(c_i|M_{e_j})$, the probability of generating the given surface form context according to each of the candidate entities' models. A very common way to do this, is to apply the multinomial unigram language model (e.g., [Bar15; Gan16; Han11a]) that ignores all the conditioning contexts and estimates each term separately. It is equivalent to the multinomial Naive Bayes model

with entities representing the classes. When using this model, we have:

$$p(c_i|M_{e_j}) = \prod_{w_k \in c_i} p(w_k|M_{e_j}) \tag{3.3}$$

To estimate $p(w_k|M_{e_j})$, the authors of [Bar15; Gan16; Han11a]) leveraged all surface form mentions in a document-centric KB, and got the maximum likelihood estimation of $p(w_k|M_{e_j})$ as follows:

$$p(w_k|M_{e_j}) = \frac{count_{e_j}(w_k)}{\sum_{w_l \in T} count_{e_j}(w_l)} \tag{3.4}$$

The function $count_{e_j}(w_k)$ returns how often word $w_k$ has been annotated in the context (context window size is set by the respective approach) of entity $e_j$. Parameter $T$ denotes the underlying term dictionary. The authors noted that a robust estimation of $p(w_k|M_{e_j})$ is often not possible and, thus, further smoothed the probabilities by applying the Jelinek-Mercer smoothing method [Jel80].

Blanco et al. [Bla15] noted that the presented model above does not take the semantic similarity of words into account. For instance, if a word $w_1$ is a describing word for entity $e_j$ and another word $w_2$ provides a high semantic similarity to $w_1$, it is very likely that $w_2$ is also a relevant word for $e_j$. To consider this, the authors first created continuous vector representations $v(w_k)$ for all words in the dictionary, specifically Word2Vec embeddings [Mik13a]. After mapping each entity $e_j$ to a vector representation $v(e_j)$, the authors modeled $p(w_k|M_{e_j})$ with a binary logistic regression classifier:

$$p(w_k|M_{e_j}) = \sigma(v(w_k)^\intercal \cdot v(e_j)) \text{ , with } \quad \sigma(x) = \frac{1}{1+e^{-x}} \tag{3.5}$$

The vectors $v(e_j)$ were trained with a $L_2$-regularized logistic regression approach to distinguish between the positive and negative training examples [Bla15]. The authors used the Yahoo Search Query Log To Entities[1] data set for training, testing and evaluation. The evaluation revealed that this approach provides strong results when linking surface forms in queries. It currently represents the state-of-the-art method in this domain.

***Topic Models***

Another sophisticated method to model the entity context are topic models, which are mainly associated with Latent Dirichlet Allocation (LDA). LDA is a Bayesian probabilistic model that describes document corpora in a fully generatively way [Ble03]. The original model of LDA assumes a fixed number $K$ of topics in a given document corpus $D$ where each document $d$ is a mixture of topics $k \in K$. In LDA, the words (i.e., observable variables) within a given document are generated as follows (the corresponding graphical model is depicted in Figure 3.2):

1 http://webscope.sandbox.yahoo.com/, last accessed on 2016-11-29

1. For each document $d$, a document-specific topic proportion is drawn $\theta_d \sim Dir(\alpha)$.
2. For each topic $k$, a distribution over all words is drawn $\phi_k \sim Dir(\beta)$.
3. For every word position $l$ in a document $d$:
   a) A topic $z_l$ is randomly chosen according to $z_l \sim Multi(\theta_d)$.
   b) A word $w_l$ is randomly chosen from topic $z_l$: $w_l \sim Multi(\phi_{z_l})$.

The parameters $\alpha$ and $\beta$ are the Dirichlet priors on the per-document topic distributions and per-topic word distributions [Kot00]. $Mult(\cdot)$ and $Dir(\cdot)$ denote a Multinomial distribution or a Dirichlet distribution, respectively. Since the inference process in LDA is intractable, either a Gibbs sampling algorithm [Por08] or a variational algorithm [Hof13] is typically employed for approximation.

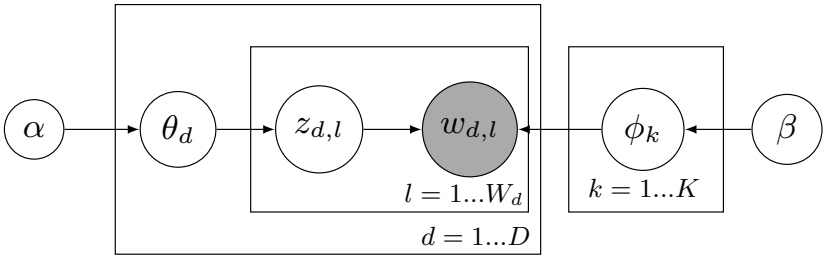



**Figure 3.2:** Graphical model for Latent Dirichlet Allocation

LDA has been leveraged for EL by several authors. For instance, Pilz and Paaß [Pil11] adapted the model to compute a surface form context - entity matching score. More specifically, they learned a topic model on a Wikipedia subset first. Next, they inferred the probability of topic $k$ for each word $w_l$ in an arbitrary document $d$ (e.g., surface form context or entity description). Further, they derived the average probability of topic $k$ for document $d$ by averaging the probabilities of topic $k$ for each word $w_l$ in $d$. Based on these probabilities, the authors yielded topic distributions for an input document (context of a surface form) and the candidate entity descriptions. To compute how well an entity fits to the surface form context, the topic distribution of the context was compared to the topic distribution of candidate entities' descriptions by applying either the Kullback-Leibler divergence [Kul51], the Jensen-Shannon divergence [Lin06] or the Hellinger distance [Ble09]. Very similar topic model approaches to leverage the context of the input document were also applied and evaluated in other works (e.g., [Liu13; Zha11a]).

The authors Houlsby and Ciaramita [Hou14; Hou13] modified the general LDA approach by associating each topic $k \in K$ directly with a single Wikipedia entity (article), resulting in $\approx 4$ million topics. The underlying words for topics were generated from article titles, the titles of all Wikipedia pages that redirect to articles, and the anchor text of all intralinks within Wikipedia (a surface form phrase also represents a 'word' in the model). Since the topics of all articles are a-priori fixed, the authors initialized the topic-word distributions $\phi_k$ by using the empirical distributions from Wikipedia counts: $p(k|w_l) = \frac{count(k,w_l)}{count(w_l)}$. Function $count(k, w_l)$ returns the number of occurrences of word $w_l$ in the Wikipedia article associated with topic $k$. Based on these assumptions, the inference step in the

approach computes the topic assignments for each word in an input document. Thus, the approach assigns an entity directly to a given surface form, simultaneously denoting the linking result. There also exist some other, even more complex topic model approaches that jointly incorporate contexts, entity types and word distances. We address these models in Section 3.2.6.

Li et al. [Li13] applied topic models to mine evidences for entities. More specifically, given a surface form $m_i$, an underlying reference knowledge base $KB$ (i.e., Wikipedia) as well as an external document corpus $C$, the task is to mine evidences in form of keywords from $KB$ and $C$ to improve EL. Table 3.3 shows an example of such evidences for a set of ambiguous entities.

**Table 3.3:** Mined evidences for the entities Michael I. Jordan, Michael B. Jordan, Justin Bieber and Owen Bieber [Li13]

| Surface Forms | Candidate Entities | Mined Evidences |
|---|---|---|
| Michael Jordan | Michael I. Jordan | layers, nonparametric, non-linear, distinguished, chen, pehong, david, marina, meila, kearns ... |
| | Michael B. Jordan | wood, oscar, role, peters, detmer, larry, true-frost, pryzbylewski, octavia, troubled, gilliard ... |
| Bieber | Justin Bieber | music, london, jail, guinness, selena, thc, technology, black, wayne, leaked, ... |
| | Owen Bieber | jobs, automobile, corporation, approved, support, vote, organizer, conventions, worker, worley ... |

Given a surface form $m_i$, the underlying incremental evidence mining algorithm first retrieves all possible candidate entities based on all annotated surface forms in $KB$. Each candidate entity is then treated as a single topic. For each occurrence of $m_i$ that is linked to an entity in $KB$, the surrounding context is extracted with each surrounding context denoting a document in the succeeding mining task. In other words, all created context documents are labeled with one specific topic (entity) instead of a topic distribution as the case in standard topic models [Ble03]. In the next step, a topic model is trained based on the surface form $m_i$ and the corresponding generated documents. Then in each mining iteration, additional unlabeled documents are added to the existing document set. It follows a topic assignment to each unlabeled document based on the underlying entity-word distribution of the model. Finally, the resulting new entity-word distribution will be adapted in the topic model. The authors used their approach to directly link a surface form to a candidate entity by interpreting a query (i.e., surface form and its describing document) as a new unlabeled document.

However, the authors also suggested to use the entity-word distributions as evidence indicator. Words with a high probability describing an entity can be seen as evidence terms and may contribute to a (significant) linking improvement. Hereby, an open problem is how to merge the different entity-word probability distributions for different surface forms $m_i$, because each surface form $m_i$ has its own topic model [Li13].

***Neural Network Models***

In contrast to the works mentioned before, He et al. [He13a] used a deep learning technique [Hin06] in order to compute a surface form context - entity similarity score. The algorithm consists of a greedy-wise pretraining stage and a supervised fine tuning stage. Figure 3.3 shows the network architecture of the underlying deep learning model.

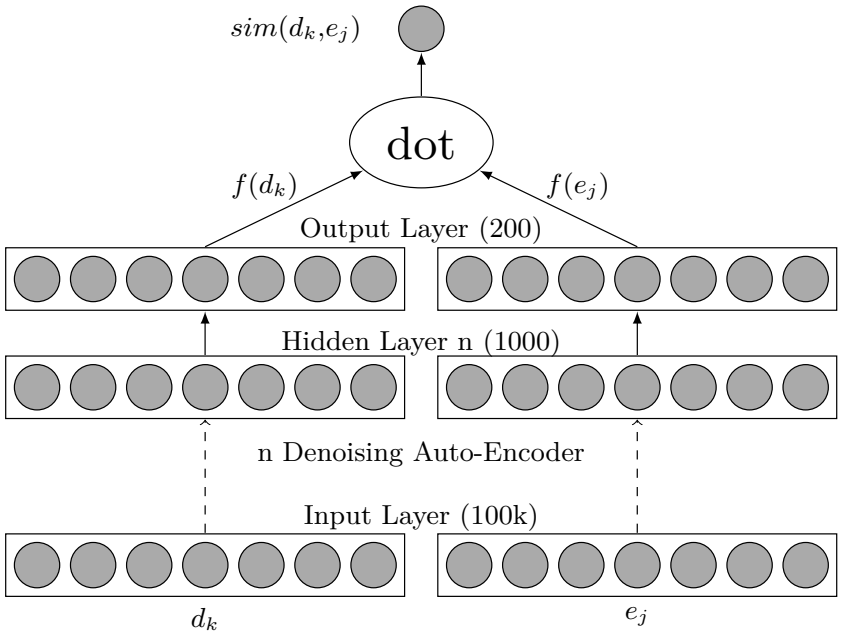



**Figure 3.3:** The deep learning architecture of the entity-context model was proposed in [He13a]. The number of layer units is given in parentheses.

Basically, the network is trained using Wikipedia paragraphs $d_k$ and the corresponding entities $e_j$ as a binary bag-of-words input. More specifically, each Wikipedia paragraph $d_k$ contains a surface form that refers to entity $e_j$. In other words, the paragraphs can be seen as surrounding context of a surface form referring to $e_j$. However, a set of denoising auto-encoders [Ben07] is stacked to explore general concepts encoding paragraph $d_k$ and entity $e_j$. In the following supervised fine-tuning stage ('Hidden Layer n' in Figure 3.3), the document and entity representations are optimized toward entity annotations located in Wikipedia. For that reason, an additional layer is stacked on top of the learned representations to capture problem specific structures (given by the training data). The representation of $d_k$ and $e_j$ after the problem-specific layer is denoted as $f(d_k)$ and $f(e_j)$. The surface form context - entity similarity is then defined as the dot product between $f(e_i)$ and $f(d_j)$. The evaluation revealed that the learned similarity achieves state-of-the-art performance on two data sets when it is integrated in a graph-based approach.

In **summary**, we state that analyzing the textual context of surface forms is an important feature in EL algorithms. No matter in which kind of documents the surface forms are located (i.e., web sites, research papers, news articles or tables), most EL approaches leverage the surface forms' textual context to improve linking accuracy. Topic models

(e.g., [Hou14; Pil11]) and deep learning approaches (e.g., [He13a]) are the current state-of-the-art techniques when enough training data is available (e.g., Wikipedia). Linking more specialized or less popular entities (i.e., entities with less textual descriptions) benefits from methods that rely on special keywords (e.g., [Hof12]) or other explicit contextual features (e.g., [Cuc07]).

### 3.2.5 Topical Coherence

Another crucial feature is the topical coherence between multiple entities within the same document. It is based on the assumption that a document largely refers to coherent entities from one or few related topics. To take advantage of topical coherence features we first have to identify additional surface forms in the document with the help of an Entity Recognition step. Very often multiple surface forms are a-priori given in order to improve the EL accuracy by collectively linking all surface forms to entities. In collective EL, topical coherence is usually modeled as a pairwise entity relatedness measure that describes the semantic relatedness between an entity pair $e_j$ and $e_k$. We note that entity relatedness, semantic relatedness and semantic similarity are often used interchangeably in the EL community. In this work, we refer to entity relatedness. However, entity relatedness is commonly defined as a real value function $\rho : \Omega \times \Omega \mapsto [0, 1]$ where 0 and 1 are the minimum and maximum relatedness values, respectively [Cec13]. Figure 3.4 shows an example entity relatedness graph, including the candidate entities of the two input surface forms 'TS' and 'New York'. The selection of the most relevant entity assignment is often implemented on top of this graph, where the edges between entities are weighted by the relatedness function $\rho$.

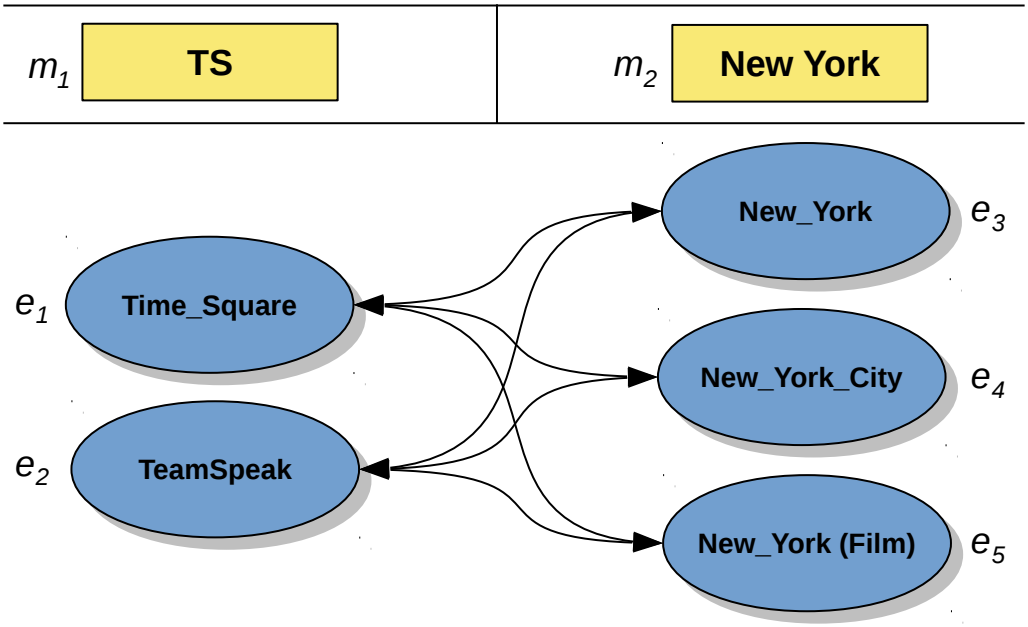



**Figure 3.4:** An entity relatedness graph with candidate entities for the surface forms 'TS' and 'New York'. For simplicity of representation the relatedness scores between entity pairs are omitted.

Most existing state-of-the-art relatedness measures rely on KB information to produce a numerical approximation of the relatedness between two entities. Since a huge number of measures were proposed in the literature, we focus on the most important ones that were explicitly evaluated in the context of EL. A survey about recent advances in methods of semantic relatedness in general can be found in [Zha13b]. In the following, we distinguish between entity relatedness measures based on document-centric KBs and entity-centric KBs.

***Document-Centric Knowledge Bases***

Document-centric KBs provide a rich source for entity relatedness measures. The leading KB for entity relatedness computation is Wikipedia, which provides a broad range of entity information to leverage. For instance, Cucerzan [Cuc07] suggested to simply analyze the agreement between categories of two candidate entities. In other words, if two entities share the same categories, then both entities are semantically related. Ponzentto and Strube [Pon07] chose a similar approach that computes the relatedness between entities based on the paths found along the category network and the word overlap of the entities' article pages. Cai et al. [Cai13] computed entity relatedness with the help of a co-occurrence matrix and argued that if two entities often co-occur within a given context window, they are semantically related. A matrix entry defines the number of co-occurrences of two entities within a given window size across all Wikipedia documents. Hence, each entry defines a relatedness score between the respective entities. A more advanced approach was proposed by Milne and Witten [Mil08a; Mil13; Mil08b]. Their Wikipedia Link-based Measure (WLM) is based on the Normalized Google Distance [Cil07] and assumes that two Wikipedia articles are related if there are many Wikipedia articles that link to both. Given two Wikipedia entities (articles) $e_j$ and $e_k$, the relatedness between both is defined as:

$$Co_{WLM}(e_j, e_k) = 1 - \frac{log(max(|U_j|,|U_k|)) - log(|U_j \cap U_k|)}{log(|WA|) - log(min(|U_j|,|U_k|))} \tag{3.6}$$

The sets $U_j$ and $U_k$ refer to the Wikipedia articles that contain links to the article pages $e_j$ and $e_k$, and $WA$ is the set of all Wikipedia articles available. Bhagavatula and Thanapon [Bha15] applied the modified version of the WLM relatedness measure by Hecht et al. [Hec12]. In this adaption, the links in the first paragraph of a Wikipedia page are considered more important than other links when computing entity relatedness. Ratinov et al. [Rat11] also adopted the WLM along with the Pointwise Mutual Information [Chu90] (PMI) relatedness measure, which is defined as follows:

$$Co_P(e_j, e_k) = \frac{|U_j \cap U_k|}{|U_j| \cdot |U_k|} \tag{3.7}$$

Furthermore, Guo et al. [Guo13] computed the Jaccard distance to measure the relatedness between two Wikipedia entities:

$$Co_J(e_j, e_k) = \frac{|U_j \cap U_k|}{|U_j \cup U_k|} \tag{3.8}$$

The proposed methods are simple but proved to be effective and performant using Wikipedia.

Other, more complex approaches were proposed recently. For instance, Ceccarelli et al. [Cec13] formalized the problem of learning entity relatedness as a Learning to Rank (LTR) problem [Liu09]. More specifically, the measure is a weighted linear combination of 27 established features like WLM, PMI, Kullback-Leibler divergence [Kul51] and Jaccard similarity between the Wikipedia in-link article sets. The results demonstrate a better entity relatedness estimation and show improvements toward other state-of-the-art approaches. Another example is the work by Guo et al. [Guo14], who created semantic signatures by creating and traversing a graph $G = (V, E)$ with $V$ denoting the set of entities and $E$ denoting the set of edges. An edge is added if two entities (i) are mentioned in the Wikipedia article within a window of 500 words, or (ii) there is an interlink between both entities on Wikipedia, i.e., the article page of one entity links to another article page. To measure the relatedness between an entity pair, the authors applied a random walk with restart [Ton06], which is a stochastic process to traverse the created graph to obtain a probability distribution for each entity. The probabilities were interpreted as relatedness scores between entities. In order to improve computation performance, the authors created a specific subgraph for each entity that only contains adjacent nodes within a given range.

A significant drawback of the mentioned approaches is the dependency of Wikipedia documents describing a specific entity. As a consequence these methods can be only applied for Wikipedia (or other document-centric KBs where each document describes a specific entity). Anyway, the LTR approach by Ceccarelli et al. [Cec13] described before and the graph-based approach by Guo et a. [Guo14] can also be applied to non-Wikipedia document-centric KBs if some features are omitted and minor adaptations are made when creating the respective entity graphs.

The document-centric KB agnostic approach by Shen et al. [She12a] leverages two categories of information. On the one hand, entity relatedness is based on the contexts where the entities appear in a document-centric KB. Hereby, they rely on an extension of the distributional hypothesis [Har54] by assuming that entities that occur in similar contexts are semantically related. In order to measure the distributional context similarity, the authors calculated the cosine similarity of the entities' n-gram vectors. The latter are created by analyzing the surrounding context of where the entities have been annotated (e.g., context of the respective surface forms). On the other hand, the authors additionally measured entity relatedness based on the type hierarchy assuming that two entities are related if they are in close places in a type hierarchy. More specifically, the type-hierarchy-based similarity is described by the similarity of the entities' type sets. To compute the similarity between two types, the authors adopted the information-theoretic method proposed in [Lin98]. Finally, the weighted sum of both similarity measures defines the entity relatedness measure. The measure was evaluated on web lists and tables, and shows strong results on the respective data sets. Overall, the authors evaluated this approach using Wikipedia as underlying KB. Nevertheless, the approach can be applied to any document-centric KB, even when an entity type hierarchy is not available (by using the corpus-based measure only).

A lot of entity relatedness measures achieve excellent results with popular entities that provide sufficient data in the KBs, but lack accuracy for long tail and newly emerging

entities. To address this drawback, Hoffart el al. [Hof12] proposed an efficient measure called KORE, a two-stage approach allowing partial matches between entity keyphrases: (i) two keyphrases are paired up if there is a partial match of at least one word, with word-level weights influencing the matching score, and (ii) a weighted Jaccard coefficient captures the overlap between the keyphrase-sets of two entities, where the scores of the partially matching keyphrase pairs are aggregated and the phrase-level weights are considered [Hof12]. To ensure fast computation, the authors approximated keyphrases by min-hash sketches [Bre72], which were then organized by locality-sensitive hashing [Gio99]. The authors evaluated their approach on three different data sets and outperformed other measures like WLM. With relying on entity keyphrases only, this approach is KB-agnostic since keyphrases can be automatically generated from different sources. A very similar approach was proposed for biomedical entities by Rybinski2014 et al. [Ryb14], who utilized keyphrases to compute a relatedness score for knowledge-poor entities.

***Entity-Centric Knowledge Bases***

Apart from document-centric KBs, entities are often described within an entity-centric KB, especially in the biomedical domain. A very simple, binary approach to measure entity relatedness in these KBs is to regard direct relations between entities. If, and only if, two entities are directly connected via relation in an entity-centric KB, the entities are related. Limaye et al. [Lim10] used these binary relations as an important feature to annotate web tables with entities, types and relations located in the YAGO KB. Another approach that utilizes binary relations was proposed by Usbeck et al. [Usb14]. Their entity-centric KB agnostic approach achieves state-of-the-art results when linking surface forms located in web documents to DBpedia entities by exclusively using DBpedia knowledge.

In entity-centric KBs, entities are mostly organized in a strict hierarchy, where it is convenient to measure entity relatedness according to structural measures that find path lengths between entities. For instance, Rada et al. [Rad89] developed a measure based on path lengths between entities in the Medical Subject Headings (MeSH) ontology. The authors exploited *broader than* relations, which successfully provide more or less specific concepts as one travels from entity to entity. In 2004, Caviedes and Cimino [Cav04] developed the *CDIst* measure that finds the shortest path between entities in the UMLS KB. Moreover, they showed that even these simple approaches capture entity relatedness satisfactorily in that KB.

A generic relatedness measure for arbitrary concepts (e.g., entities) located in biomedical ontologies was presented by Ferreira and Couto [Fer11]. Their measure depends on the relevance of one concept to another one and the neighborhood of the underlying concepts. A relevance factor $\omega(x \rightarrow y)$ expresses the relevance of concept $x$ with relation to concept $y$ and a neighborhood $N(x)$ defines the adjacent concepts of concept $x$. The authors emphasized that both, the neighborhood $N(x)$ of a concept and the relevance factor $\omega(x \rightarrow y)$, can be adapted to a wide number of situations. For instance, $N(x)$ can be defined as the set of concepts that are connected to concept $x$ with at most $M$ relations. Moreover, authors suggested to compute the relevance factor $\omega(x \rightarrow y)$ based on the relationship types of the path from $x$ to $y$. For more detailed information and suggestions about relevance factor computation, we refer to the original work [Fer11]. Finally, the relatedness between the

concepts $x$ and $y$ is measured through the overlap in their neighborhood:

$$rel(x,y) = \frac{\sum_{k \in N(x) \cap N(y)} \omega(k \to x) + \omega(k \to y)}{\sum_{k \in N(x) \cup N(y)} \omega(k \to x) + \omega(k \to y)} \tag{3.9}$$

with $\omega(x \to y) = 0$ if $x \notin N(y)$. Since the presented relatedness measure can be applied to arbitrary concepts in an ontology, it is well suited for various ontologies in the (biomedical) domain.

Another, more complex approach proposed by Hulpus et al. [Hul15] strictly interprets an entity-centric KB as graph $G = (V, E)$ and analyzes the graph distance between two concepts (e.g., entities). The authors main rationale was that a relation between two arbitrary concepts in the entity-centric KB is stronger if each of the concepts is related through the same type of relation to fewer other concepts [Hul15]. This has been defined as *exclusivity* of relations. More formally, given a set $\mathcal{T}$ of edge types, the *exclusivity* of an edge with type $\tau \in \mathcal{T}$ that links the concepts $x$ and $y$ is defined as follows:

$$exclusivity(x \xrightarrow{\tau} y) = \frac{1}{|x \xrightarrow{\tau} *| + |* \xrightarrow{\tau} y| - 1} \tag{3.10}$$

Thereby $|x \xrightarrow{\tau} *|$ denotes the number of relations of type $\tau \in \mathcal{T}$ that exit node $x$, and $|* \xrightarrow{\tau} y|$ denotes the number of relations of type $\tau \in \mathcal{T}$ that enter node $y$ [Hul15]. Additionally, given the graph $G$ (entity-centric KB), a path $P$ through $G$ is defined as $P = n_1 \xrightarrow{\tau_1} n_2 \xrightarrow{\tau_2}$ ,...,$n_L$ with $\tau_l \in \mathcal{T}$ and $n_l \in V$. The weight of an arbitrary path is then given by

$$weight(P) = \frac{1}{\sum_l 1/exclusivity(n_l \xrightarrow{\tau_l} n_{l+1})} \tag{3.11}$$

Given these definitions, the relatedness between two concepts $x$ and $y$ is computed by summing up the path weights of the top-$k$ paths that show the highest weights:

$$rel^{(k)}_{(x,y)} = \sum_{P_h \in P^{(k)}_{x,y}} \alpha^{length(P_h)} weight(P_h) \tag{3.12}$$

with $P^{(k)}_{x,y}$ denoting the set of the top-$k$ weighted paths between $x$ and $y$ and $\alpha$ denoting a path length decay factor to preference shorter paths with $\alpha \in [0,1]$. The measure was evaluated on DBpedia only but outperforms other related state-of-the-art entity relatedness measures on this KB.

Recently, Huang et al. [Hua15] leveraged deep neural networks (DNN) [Hin06; Ian16] to measure entity relatedness in entity-centric KBs. In their novel deep semantic relatedness model (DSRM), for each entity $e_j$, the authors incorporated connected entities $E_j$ (entities connected via relations), relations $R_j$, entity types $ET_j$ and entity descriptions $D_j$. The model learns latent semantic entity representations that capture the semantics of entities. To learn these entity representations, the authors encoded various semantic knowledge from entity-centric KBs into a DNN. Figure 3.5 shows the architecture of the DSRM.

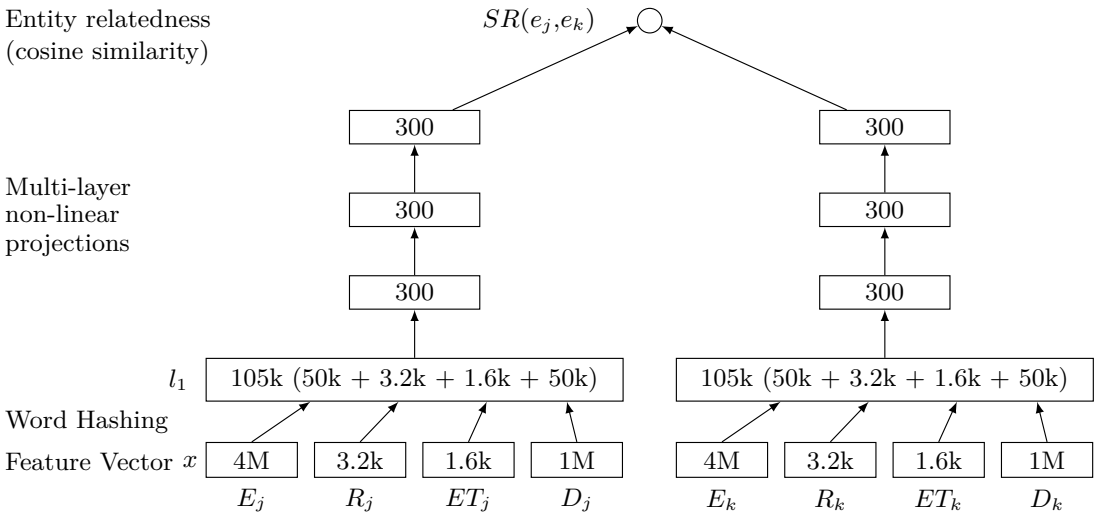



**Figure 3.5:** Architecture of a deep semantic relatedness model [Hua15]. The numbers in rectangles denote the size of the feature vectors.

First, a letter n-gram-based word hashing technique [Hua13] is applied to reduce the dimensionality of the bag-of-word term vectors of the input entity knowledge. Second, on top of the word hashing layer, the DSRM has multiple hidden layers to perform non-linear transformations. This allows the DNN to learn semantic features with respect to a self-defined objective function designed for the entity relatedness task. Finally, the semantic representation of an entity $e_j$ is obtained from the top layer. Given the semantic representations of two entities, the cosine similarity is used to calculate its relatedness. To train the DNN, the authors used relations in the underlying entity-centric KB as positive training examples and automatically generated negative training examples from Wikipedia. The approach achieves state-of-the-art results on two data sets, but, unfortunately, was not evaluated on other data sets to confirm the results.

As already mentioned, a major drawback of many relatedness measures is the dependency or optimization on a specific knowledge source. To overcome this deficit, Han and Zhao [Han10] proposed the Structural Semantic Relatedness (SSR) measure, which captures the knowledge from different knowledge sources. In contrast to other works that compute the relatedness directly between two entities, the authors computed the relatedness between two surface forms. For that purpose, the authors used Wikipedia, WordNet and a named entity co-occurrence corpus as concept databases. Moreover, they suggested to extract all concepts in the surface forms' contexts and represent them as the nodes in a unified semantic graph. Two concepts in the graph are connected if these concepts are (strongly) related within one of the used KBs. To score the degree of relation and to weight the edges in the semantic graph, the authors used the WLM relatedness [Mil08a] for Wikipedia concepts, the semantic similarity measure proposed in [Lin98] for WordNet concepts and the Google Similarity Distance [Lin98] for the named entity corpus. To exploit the graph structure and the edge weights in the graph, the authors extended the measure proposed in Leicht et al. [Lei05] to measure the structural semantic relatedness between the extracted

concepts around the surface forms. Finally, to measure the similarity between two surface forms, it was suggested to represent each surface form as a weighted vector of its extracted concepts. Hence, a surface form $m_i$ is represented as a context vector $v_i = w_{i1},w_{i2},...,w_{iL}$, where $w_{ik}$ is the $k$-th concept weight of surface form $m_i$ using the TF-IDF weight. The semantic similarity between two surface forms is then computed as:

$$sim(m_i, m_h) = \frac{\sum_l \sum_k w_{il} w_{hk} S(l,k)}{\sum_l \sum_k w_{il} w_{hk}} \tag{3.13}$$

which is the weighted average across all structural semantic relatedness values $S(l,k)$ of the surface forms' concepts [Han10]. Given the relatedness between all surface forms, we are able to cluster the surface forms that are topically related. This allows us to collectively link the entities cluster-wise since assigned entities within a cluster should be highly topically coherent.

In **summary**, we state that measuring the relatedness between entities is an essential step in collective EL approaches. Most existing relatedness measures were aligned to and evaluated on specific KBs but lack robustness in terms of other KBs, in particular structurally different KBs. Regarding document-centric KBs, entity keyword comparisons or a combination of different measures (e.g., WLM [Mil08a], Pointwise Mutual Information, Jaccard distance) showed to be very effective. In contrast, various graph-analysis techniques were applied for entity-centric KBs and achieved satisfying results. However, we assume that entity relatedness measures based on deep learning techniques, as proposed in [Hua13], can further improve collective EL.

### 3.2.6 Joint Feature Modeling

In the following, we present approaches that model different aspects of EL within a joint model. Instead of **computing and combining** distinct feature values or distributions in an EL algorithm, these approaches jointly combine multiple aspects, as proposed in the sections before, in a single model. A typical and well-known technique to jointly model these features are topic models, like LDA. A brief description of LDA and the respective notations can be found in Section 3.2.4 on Page 35.

One of these models was proposed by Kataria et al. [Kat11], who learned a semi-supervised hierarchical topic model called *Semi-supervised Wikipedia-based Pachinko Allocation Model* (WPAM). The model captures the rich textual descriptions of entities and their category hierarchy in Wikipedia. In addition to each entity defining a specific topic in the model, the authors made the following two crucial extensions:

- **Wikipedia-based Pachinko Allocation Model:** With being an extension of the Pachinko Allocation Model for LDA [Li06], the model allows to additionally capture topic correlations within documents, thus enabling collective EL. In contrast to the original Pachinko Allocation Model that focuses on a fixed four-level topic hierarchy, WPAM leverages the entire Wikipedia category hierarchy. The category hierarchy represents a directed acyclic graph structure and groups semantically related entities into relevant categories.

- **Supervision:** The authors integrated a form of weak supervision into the standard LDA model by leveraging and integrating Wikipedia annotations (i.e., annotated surface forms) into their system to improve linking accuracy. The key idea was to bias the topic-word distribution $\phi_k$ in favor of surface forms (words) that were often annotated with topic/entity $k$ and to bias the document-topic distributions $\theta_d$ in favor of topics that were referred by the surface form annotations within $d$.

In the underlying evaluation, the WPAM approach was (slightly) superior to other standard LDA approaches for EL. A detailed overview of the generative model can be found in the respective work [Kat11].

Another approach to model the textual context and the topical coherence with topic models in Wikipedia was proposed by Sen [Sen12], namely *Collective context-aware topic models* (CA). In contrast to Kataria et al. [Kat11], the authors of this approach did not leverage the Wikipedia category system. Instead, they proposed a separate topic model to learn groups of entities based on a document-centric KB like Wikipedia. Each group represents a Multinomial distribution over entities and describes the entities' topical coherence with respect to this group. A major issue in generating entity groups was the optimal number of groups given a specific corpus to achieve the best EL results. However, in addition to entity groups that model topical coherence, the authors incorporated word proximity in their model. It is based on the idea that words that appear in the context of an entity are more likely to be associated with this entity. In contrast to LDA, where each word $w_h$ in a document $d$ is generated independently, the CA model generates a document $d$ as a sequence. This means that generating a word $w_h$ also depends on the previous annotated word or words in a previously annotated sentence or paragraph. A thorough evaluation of differently modified topic-models and the effects of differently sized entity groups showed that the proximity of words to entities as well as modeling topical coherence significantly contribute to a high EL accuracy [Sen12]. This topic model is also applicable to other document-centric KBs and does not depend on Wikipedia-specific features.

The current state-of-the-art topic model for EL on the well-known IITB data set [Kul09] was proposed by Han et al. [Han12] in 2012. The model also incorporates topical coherence and is based on three types of global knowledge, namely: Topic knowledge $\phi$, entity name knowledge $\psi$ and entity context knowledge $\xi$. The topic knowledge describes that each entity $e_j$ in a document $d$ is generated based on a topic $z_l$, with $z_l$ containing semantically coherent entities (similar to the groups in [Sen12]). Each topic is modeled as Multinomial distribution of entities with the probability denoting the likelihood of an entity $e_j$ getting extracted from topic $z_l$. The entity name knowledge describes that a surface form $m_i$ is generated based on all possible annotations of the underlying entity. Hence, the name knowledge of an entity $e_j$ is modeled as a Multinomial distribution of its surface form annotations in the overall document corpus $D$. Finally, entity context knowledge describes that all words $w_n$ are generated using its context knowledge. In other words, the context knowledge of an entity $e_j$ is modeled as a Multinomial distribution of words, with the probability describing the likelihood of $w_n$ occurring in the context of $e_j$. Given the topic knowledge $\phi$, entity name knowledge $\psi$ and entity context knowledge $\xi$, the generative process can be described as follows [Han12]:

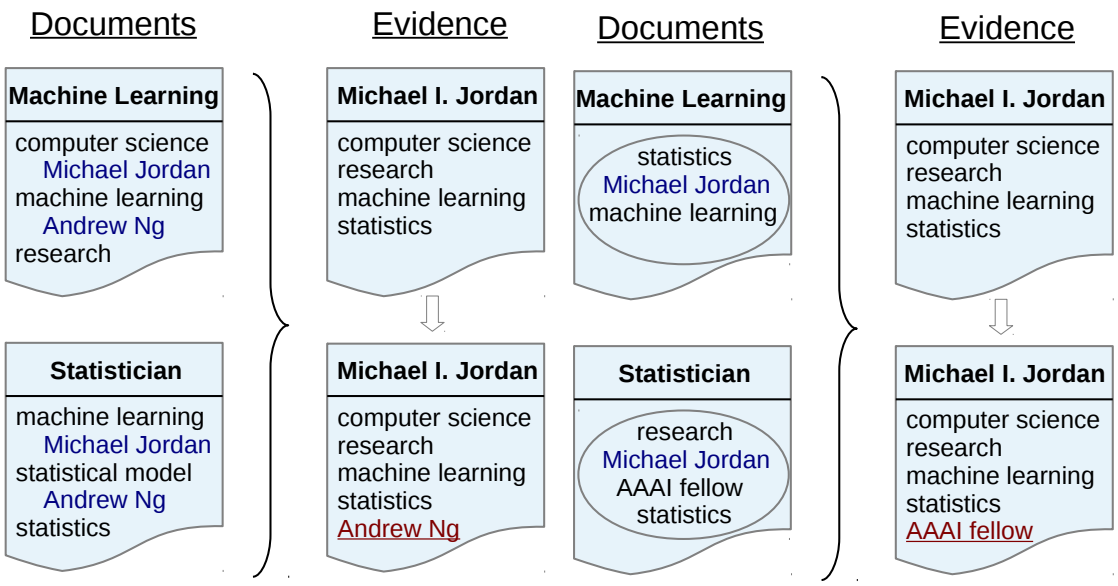
1. For each document $d \in D$, sample the topic distribution $\theta_d \sim Dir(\alpha)$
2. For each surface form position $i$ in document $d$:
   a) Sample a topic assignment $z_i \sim Mult(\theta_d)$
   b) Sample an entity assignment $e_i \sim Mult(\phi_{z_i})$
   c) Sample a surface form $m_i \sim Mult(\psi_{e_i})$
3. For each word position $l$ in document $d$:
   a) Sample a target entity from $d$'s referent entities $a_l \sim Uniform(e_{m_1}, e_{m_2},...,e_{m_d})$
   b) Sample a describing word using $a_l$'s context word distribution $w_l \sim Mult(\xi_{a_l})$

The global knowledge $\phi$, $\psi$ and $\xi$ is not a-priori given. Hence, the authors estimated $\phi$, $\psi$ and $\xi$ through Baysian Inference by integrating the knowledge generation process into the topic model. The authors determined the best number of topics empirically, resulting in $K = 300$.

A recently proposed topic model approach by Li et al. [Li16] links entities defined in linkless KBs. The approach is based on the preceding 'Evidence Mining' work of [Li13] (proposed in Section 3.2.4). Linkless KBs are a special case of document-centric KBs. More specifically, a linkless KB comprises a set of isolated documents $D$ with each document $d_j \in D$ describing an entity $e_j$. Cross-document or intra-document hyperlinks are not necessarily required within the documents in $D$. While other topic model approaches generate one model for the entire KB and, hence, each entity is described through its own topic, this approach generates a small topic model for each **unique** surface form $m_i$ using a small subset of the KB. More specifically, for each surface form $m_i \in \mathcal{M}$, with $\mathcal{M}$ denoting the set of surface form strings, a set of candidate documents (i.e., the documents of candidate entities) and a set of surface form documents (i.e., documents that contain the same or very similar surface forms as $m_i$) are extracted. These documents are unified to a document set $D_{m_i}$ for surface form $m_i$. The authors modeled each of the candidate entities as a single topic in $D_{m_i}$, combined with some additional, artificial topics for background words and general topics within the documents. Further, the model tries to mine additional word evidences using the set of surface form documents by mimicking the following effects of cross-document hyperlinks [Li16]:

- **Semantic Relatedness:** Generally, two entities $e_1$ and $e_2$ are related if they share the same source entities of incoming hyperlinks. Without hyperlinks, the topic model captures the relatedness by adding $e_1$'s and $e_2$'s names into each others word evidences. For instance, as shown in Figure 3.6(a), the entities *Michael I. Jordan* and *Andrew Ng* are semantically related, both co-occurring in many documents. Additionally, words like 'research' and 'machine learning' that appear in *Michael I. Jordan*'s entity description also appear in these documents. While these words are supporting evidence for *Michael I. Jordan*, we can also associate 'Andrew Ng' as *Michael I. Jordan* 's evidence, since 'Andrew Ng' co-occurs with *Michael I. Jordan*'s representative words.

- **Description Expansion for Context Similarity**: If an entity $e_1$ is linked in the document of entity $e_2$ with mention $m_i$, then the surrounding context of $m_i$ may contain additional evidence words for $e_1$. Despite non-existing hyperlinks, this approach is able to generate such evidences by directly mining them from $D$. Figure 3.6(b) shows an example where the important descriptive word 'AAAI fellow' of entity *Michael I. Jordan* is extracted from a document containing a term referring to *Michael I. Jordan*. In our case without hyperlinks, we leverage the entity describing words like 'research' of *Michael I. Jordan* in the context to associate the term 'AAAI fellow' with the entity.




**(a)** Semantic Relatedness **(b)** Description Expansion

**Figure 3.6:** Examples of mining evidences from surface form documents [Li16]. Blue highlighted terms are referring to other entities. Red highlighted terms denote additionally mined evidence words. A circle denotes the context of a surface form within a document.

Since the overall generative process takes a considerable amount of space, we refer the interested reader to the original paper [Li16] for more details.

Another significant approach proposed by Francis-Landau [Fra16] does not unify all features within one model, but computes each feature with the same technique, namely Convolutional neural networks [LeC98] (CNN). Overall, for each of the three textual granularities in the input document (i.e., the surface form, the surface form's surrounding context and the entire input document) and two textual granularities of a candidate entity (i.e., the entity name and the entity description), the authors produced vector representations with CNNs. For this purpose, each word is first embedded into a $d$-dimensional vector space using Word2Vec [Mik13a]. Next, the authors mapped the words of each granularity into a fixed-size vector using a Convolutional network, put the result through a rectified linear unit and combined the results with sum pooling producing a

representative topic vector for each granularity. The similarity between any granularity pair is denoted as the cosine similarity between the respective topic vectors. Unfortunately, the authors omitted a couple of crucial computation and CNN details, which complicates replicability. However, by using these features in a collective EL approach, the authors achieved state-of-the-art results on two data sets [Fra16].

To **summarize**, joint features model multiple features in a unified way. The most popular approaches are topic models that collectively integrate the textual context and topical coherence between entities or surface forms. Most of these models were evaluated on data sets that provide a significant amount of textual context before and after the respective surface forms. It remains the question how these approaches perform on shorter documents like tables and tweets.

## 3.3 Disambiguation Algorithms

In Section 3.1, we presented methods to generate a set of candidate entities $\Omega_i$ for a surface form $m_i$. Since the size of the candidate entity set $\Omega_i$ is very often larger than one, the remaining problem is how to leverage and incorporate features (an overview of features is given in Section 3.2) to rank candidate entities according to their relevance. The highest ranked entity typically represents the disambiguated entity for a given surface form. These disambiguation approaches can be roughly divided into the following two categories:

- **Entity-Independent Approaches:** These approaches assume that surface forms within documents are not topically coherent, thus do not leverage relations between the surface forms. For that reason, surface forms are disambiguated separately and independently while relying on local features only (e.g., candidate entity name, popularity, textual context). An overview of these features can be found in Section 3.2.
- **Collective Approaches:** These approaches assume that surface forms within documents refer to a common topic and, hence, entity assignments to surface forms are interdependent to each other. Collective EL algorithms often leverage a combination of local and global features to incorporate the local compatibility of candidate entities as well as the topical coherence across the entire document. Features to compute the topical coherence between entities were proposed in Section 3.2.5.

To provide a more fine-grained division, we classify the candidate entity ranking modules into Vector Space Model approaches (Section 3.3.1), Information Retrieval approaches (Section 3.3.2), LTR approaches (Section 3.3.3), graph-based approaches (Section 3.3.4), probabilistic approaches (Section 3.3.5), classification approaches (Section 3.3.6) and ensemble approaches (Section 3.3.7). In the remainder of this section, we provide an overview of the respective approaches and focus on the main idea how the most appropriate target entities are determined. We note that due to a huge number of EL approaches and works in the literature, not all methods can be mentioned. Most approaches link surface forms to Wikipedia entities. For that reason, we emphasize the type of entities if the authors focus on disambiguating non-Wikipedia entities. Additionally, we note that the quality (accuracy) of the main candidate ranking algorithm is basically hard to assess since the underlying feature set plays an evenly important role.

### 3.3.1 Vector Space Model Approaches

The unsupervised Vector Space Model (VSM) is a very popular, algebraic model for representing textual documents as vectors [Man08]. In the context of EL, VSMs are used as independent ranking methods by computing the similarity between a vector representation of a candidate entity and a vector representation of a surface form. In the following, the candidate entities are ranked according to the respective similarity scores. VSM-based approaches mainly differ in the used feature set for vectorial representations as well as vector similarity computation.

A very simple VSM-based approach by Chen et al. [Che10] generates vectors for surface forms and candidate entities by using the bag-of-words of the surface form context and the candidate entity description. In addition, attributes like the surface form itself and the label of candidate entities are appended to the vector. Finally, the authors denote the surface form - candidate entity similarity as the cosine similarity of the corresponding TF-IDF weighted vectors.

A very popular approach that uses the VSM is the *DBpedia Spotlight* framework, initially proposed by Mendes et al. [Men11]. It links surface forms to DBpedia entities and is similar to the approach by Chen et al. [Che10] with utilizing the bag-of-words of surface form contexts and DBpedia entity descriptions to formulate the respective vectors. Instead of using the TF-IDF weight for weighting the vector components, the authors introduced the TF-ICF weight. Here, the Inverse Document Frequency (IDF) is replaced with the Inverse Candidate Frequency (ICF). The authors argued that IDF fails to capture the importance of a word for disambiguation. For instance, let us assume that the term 'U.S.A' occurs in only three entity descriptions out of 1 million entities overall. Further, we suppose that all three entities are generated as candidates for a specific surface form (e.g., surface form 'Washington'). Despite the word 'U.S.A' providing a rather high IDF value, it has no crucial role in the disambiguation process. For that reason, the authors introduced the ICF of a word $w_k$ to weigh words based on the ability to distinguish between candidate entities [Men11]:

$$ICF(w_k) = log\frac{|\Omega_i|}{n(w_k)} = log|\Omega_i| - log\ n(w_k) \tag{3.14}$$

$\Omega_i$ denotes the set of candidate entities for surface form $m_i$ and function $n(w_k)$ returns the number of candidate entities whose descriptions contain the word $w_k$. The theoretic explanation of the proposed ICF approach is based on the Information Theory model for queries [Den09; Sha51].

Cucerzan [Cuc07] linked entities with a collective VSM approach. Each entity describing vector represents a binary vector that is composed of two subvectors. The first subvector comprises a binary entry for each word in Wikipedia, with an entry being 'true' if the word appears in the respective entity describing article. The second subvector, however, comprises a binary entry for each existing Wikipedia category, with an entry being 'true' if the respective entity is associated with the respective category. The query vector (vector for an input document) comprises the number of word occurrences in the input document with respect to all words in Wikipedia. To account for all possible disambiguations of the surface

forms in the given document, Cucerzan extended the query vector by an additional vector that contains the number of annotated categories of all candidate entities across all surface forms. Finally, the EL systems aims to find an assignment of entities to surface forms that maximizes the context and category agreement between the annotated entities [Cuc07].

Han and Zhao [Han09] proposed a hybrid, non-collective VSM-based approach to link entities to Wikipedia. First, the authors defined the surface form - candidate entity similarity as the cosine similarity between the corresponding word context vectors weighted with TF-IDF.

Second, the approach extracts Wikipedia entities from the surrounding context of each surface form and from the description of each candidate entity using the approach in [Mil08b]. We refer to $\pi_i$ as the set of entities extracted from the context of surface form $m_i$ and refer to $\pi_j$ as the set of entities extracted from the description of entity $e_j$. Then, the Wikipedia similarity between surface form $m_i$ and candidate entity $e_j$ is computed as follows:

$$sim_{\text{Wiki}}(m_i, e_j) = \frac{\sum_{e_k \in \pi_i} \sum_{e_h \in \pi_j} w(e_k, \pi_i) w(e_h, \pi_j) r(e_k, e_h)}{\sum_{e_k \in \pi_i} \sum_{e_h \in \pi_j} w(e_k, \pi_i) w(e_h, \pi_j)} \tag{3.15}$$

which is the weighted average of all entity relatedness scores between surface form $m_i$ and candidate entity $e_j$ [Han09]. The semantic similarity $r(e_k, e_h)$ between two Wikipedia entities is the WLM semantic similarity as described in Section 3.2.5 and in the work [Mil08a]. The weights $w(e_k, \pi_i)$ or $w(e_h, \pi_j)$ help to select useful entities and describe the average relatedness of an entity and an entity set:

$$w(\mathrm{e}_k, \pi) = \frac{\sum_{e_l \in \pi} r(e_k, e_l)}{|\pi|} \tag{3.16}$$

Finally, after computing both similarities (i.e., bag-of-words-based similarity and Wikipedia concept similarity), the authors derived a hybrid similarity by summing up the weighted similarity values.

In **summary**, we state that VSMs were often employed in older EL approaches. Meanwhile, more sophisticated and supervised methods have evolved that (significantly) outperform VSM-based approaches.

### 3.3.2 Information Retrieval Approaches

Another unsupervised method to rank candidate entities are Information Retrieval approaches, which can be described as follows (cf. Figure 3.7): A system typically maintains a set of $|\Omega|$ entity representations $D = \{d_1,...,d_{|\Omega|}\}$, with $d_j$ representing $e_j \in \Omega$. In the following, a surface form $m_i$ and its surrounding (textual) context $c_i$ are used to create a query $q_i$. Given this query, the EL system retrieves entity describing documents (i.e., entity representations) that contain all or a subset of the query words, ranks the documents according to their query relevance and returns the top-$n$ ranked entity representations $\{d^i_{j,1},...,d^i_{k,n_i}\}$ for query $q_i$. Traditionally, the ranking is performed by a ranking function $f(q_i, d_j)$, with $q_i$ denoting the input query for surface form $m_i$ and $d_j$ denoting the entity representation (document).

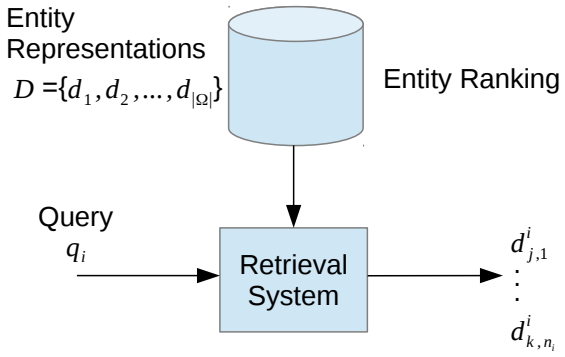



**Figure 3.7:** Standard Entity Retrieval

The current state-of-the-art EL approach using an Information-Retrieval-based method was proposed by Gottipatti and Jiang [Got11]. The authors adopted the Kullback–Leibler divergence retrieval model, a statistical-language-based retrieval model proposed by Lafferty and Zhai [Zha01]. Given a query $q_i$ and a KB entity representation $d_j$, the score is based on the Kullback-Leibler divergence [Kul51]:

$$f(q_i, d_j) = -Div(M_{q_i}||M_{e_j}) = -\sum_{w_l \in T} p(w_l|M_{q_i})\ log\frac{p(w_l|M_{q_i})}{p(w_l|M_{e_j})} \tag{3.17}$$

$M_{q_i}$ denotes the query language model for a surface form $m_i$, and $M_{e_j}$ denotes the KB entry language model for candidate entity $e_j$. Both language models are Multinomial distributions over words $w_l$ in the vocabulary $T$. Each entry language model $M_{e_j}$ is estimated via maximum likelihood estimation with Dirichlet smoothing [Zha04]:

$$p(w_l|M_{e_j}) = \frac{c(w_l, d_j) + \mu p(w_l|M_C)}{|d_j| + \mu} \tag{3.18}$$

with $c(w_l, d_j)$ retrieving the number of occurrences of word $w_l$ in $d_j$. Further, $M_C$ is a background language model across all words in the KB and $\mu$ denotes the Dirichlet prior.

To estimate the query language model $M_{q_i}$, Gottipatti and Jiang [Got11] used an empirical query word distribution:

$$p(w_l|M_{q_i}) = \frac{c(w_l, m_i)}{|m_i|} \tag{3.19}$$

The function $c(w_l, m_i)$ counts how often word $w_l$ appears in the surface form $m_i$ and $|m_i|$ denotes the number of words in surface form $m_i$. Finally, the disambiguated entity is selected according to the highest score between query and a candidate entity.

Other approaches use Apache Lucene[1] to index the Wikipedia article text of all candidate entities [Var10; Var09; Zha10b]. Since Apache Lucene is mainly based on the VSM with TF-IDF weights, these EL approaches are very similar to those proposed in Section 3.3.1 using the VSM. Nemesky et al. [Nem10], however, employed their own search engine SZTAKI [Dar08] and performed experiments when endowing the search engine with entity names, entity names and infoboxes, and finally the entire Wikipedia article.

All proposed Information Retrieval methods rank entities independently. Their application is limited since no training data is used to learn a ranking function and, thus, the ranking model has limited evidence about the most relevant candidate entities. In the following Section 3.3.3, we propose LTR methods which resemble Information Retrieval methods but leverage underlying training data to learn a ranking function $f(q_i, d_j)$.

In **summary**, we state that Information Retrieval methods for EL are not widely used. Similar to VSM approaches, training data in form of annotated documents is not needed. The underlying approaches rely on extensive entity representations (e.g., entity descriptions) in order to provide enough evidence for accurate EL results.

### 3.3.3 Learning to Rank Approaches

So far, our ranking function $f$ that is used in Information Retrieval methods (cf. Section 3.3.2) does not consider training data. The goal of LTR approaches is to construct the ranking model by means of training data. More specifically, LTR is a supervised model containing training and test phases (cf. Figure 3.8). In the following, we apply the notation introduced in Section 3.3.2. During the training phase, each query $q_i$ is associated with a number of entities (i.e., entity representations) $d_j^i$. The relevance of the documents given a query is also known. In the context of EL, relevance typically means whether an entity document is the correct result (e.g., 'positive') or not (e.g., 'negative').

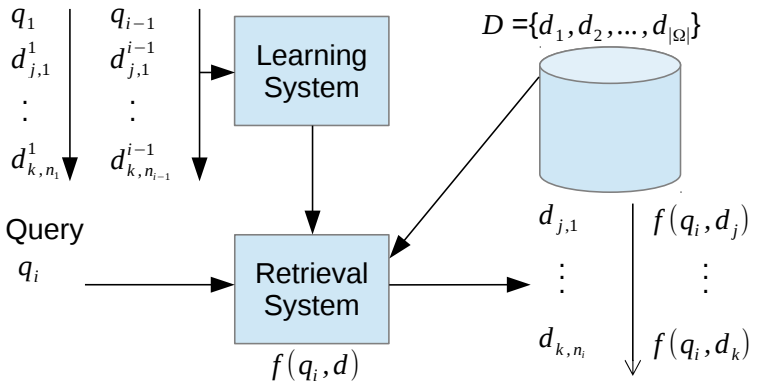



**Figure 3.8:** Learning to Rank for Entity Retrieval

1 http://lucene.apache.org/, last accessed on 2016-11-29

A LTR approach aims to train a ranking model $f(q_i, d_j)$ that assigns a score to a given query - entity representation pair $q_i$ and $d_j$, or equivalently to a given feature vector $\phi$ [Li11]. Feature definition and selection is part of a feature engineering step and depends on the respective works. However, the score between a query $q_i$ and an entity representation $d_j$ is the linear combination of the weighted feature set $\phi(q_i, d_j)$:

$$f(q_i, d_j) = w^\intercal \phi(q_i, d_j) \tag{3.20}$$

with $w$ denoting the weight vector for the respective feature set that is learned by the LTR model. Overall, three popular LTR approaches have been evolved, namely a pointwise, a pairwise and a listwise approach. An in-depth summary of the approaches can be found in [Li11; Liu09].

Nearly all EL approaches that employ a LTR approach rely on the pairwise Ranking SVM approach [Her00; Joa02]. SVM Rank uses a max-margin technique based on the training set. Given a ground truth entity assignment $e^{m_i} \in \Omega$ for a surface form $m_i$, then the score for the correct entity assignment $e^{m_i}$ should be higher than the score of all other entities $e_j \in \Omega$ with a specific margin, with $e_j \neq e^{m_i}$. The learning of the Rank SVM model can be formalized as a quadratic programming problem [Li11; She15]:

$$\min_{w, \xi_{m_i,j}} \frac{1}{2}||w||^2 + Y \sum_{m_i,j} \xi_{m_i,j} \tag{3.21}$$

$$\text{s.t.} \quad \forall m_i, \forall e_j \neq e^{m_i} \in \Omega : \langle w, x_i^{e^{m_i}} - x_i^{e_j} \rangle \geq 1 - \xi_{m_i,j}$$
$$\xi_{m_i,j} \geq 0$$

where $w$ is the weight vector, $x_i^{e_j}$ is the feature vector of entity $e_j$ for surface form $m_i$, $\xi_{m,i}$ is a slack variable, $||\cdot||$ denotes the $L_2$ norm and $Y > 0$ is the tradeoff parameter between margin size and training error [She15].

In the literature, several works employ the Rank SVM model to rank candidate entities using local features (e.g., [Dre10; Zha11a; Zha11b]). Typical groups of features are surface form matching features, popularity features and textual context features. The respective approaches mainly differ in feature selection and feature number. Bunescu and Pasca [Bun06] additionally incorporated a *taxonomy kernel* into their feature set, which allows to match the categories of candidate entities with the surrounding context of surface forms.

However, the previous approaches do not consider the topical coherence between the candidate entities across all surface forms within a document. For that purpose, Kulkarni et al. [Kul09] ranked the candidate entities for each surface form using a local feature set first. Next, given a local score, the authors additionally added a global feature to incorporate topical coherence between all candidate entities across all surface forms and solved the optimization problem with a graphical model. Several other EL approaches directly consider topical coherence in their LTR candidate entity ranking model [Rat11; She12a; She12b]. To simplify the optimization problem, the authors adopted a robust strategy: The sum of all relatedness scores between candidate entity $e_j$ of surface form $m_i$

and the top ranked candidate entity of another surface form $m_k$ (using local features only) defines the global feature:

$$Coh_{e_j,m_i} = \sum_{m_k, m_k \neq m_i} sim(e_j, top(m_k)) \tag{3.22}$$

with function $top(m_k)$ returning the most likely candidate entity for surface form $m_k$ according to a (trained) local feature function. The authors of the EL framework LINDEN [She12b] chose the WLM [Mil08a] as entity relatedness measure and determined the most likely entity of other surface forms by computing the Sense Prior $p(e_j|m_i)$. Overall, the authors learned the weights of three local features (i.e., Sense Prior and two graph-based features) and one global feature to rank candidate entities. The EL system *GLOW* by Ratinov et al. [Rat11] works similar. Apart from another feature set, the authors first trained a local linking system (using local features only) whose scores are used to select the most relevant candidate entities (cf. Equation 3.22). Another similar approach was proposed by Shen et al. [She12a], who used a temporal linking score to select the most likely candidate entity of other surface forms. In contrast to [She12b] and [Rat11], the most likely candidate entities of the other surface forms change over time, since the temporal linking score already includes global features. Instead of computing Equation 3.20 to rank each candidate entity once, the authors determined the best candidate entity combination across all surface forms iteratively by updating the temporal scores. A more sophisticated approach was proposed by He et al. [He13b], whose approach is based on stacking. Stacked generalization [Wol92] is a powerful meta learning algorithm that uses two levels of learners. Very similar to Ratinov et al. [Rat11], the authors first ranked the candidate entities of a surface form with a LTR approach using local features. In the second step, they integrated another LTR ranking step containing global features. Here, semantically related entities of the currently selected candidate entity for surface form $m_i$ are searched across ALL candidate entities of other surface forms and considered during global feature computation. In addition to the global features, the second LTR step also receives the original local features as input.

Zheng et al. [Zhe10] investigated and compared the pairwise LTR framework Ranking Perceptron [She05] to the listwise approach ListNet [Cao07]. As a result, the authors reported superior results of ListNet compared to Ranking Perceptron in the context of EL. Unfortunately, global features were omitted and the feature set exclusively comprised mundane features like surface form and textual context comparisons.

We **summarize** that LTR approaches for EL were successfully employed. The respective approaches show a strong EL accuracy if enough training data is available. One problem in these approaches has been the incorporation of global features for collective EL. In the next Section 3.3.4, we provide an overview of graph-based approaches that are better suited for collective EL.

### 3.3.4 Graph-Based Approaches

Graph-based approaches are particularly well-suited to model the interdependence of entities within collective EL approaches. Given a set of candidate entities across several

surface forms within an input document, graph-based approaches model a graph $G = (V, E)$, commonly integrating all candidate entities as nodes. Depending on the graph definitions, nodes might also represent surface forms. Further, graph edges typically denote relations between a candidate entity pair or a surface form - candidate entity pair. Figure 3.9 shows an example graph containing surface forms, corresponding candidate entities and relations (the transition weights are not normalized and are for illustrative purposes only).

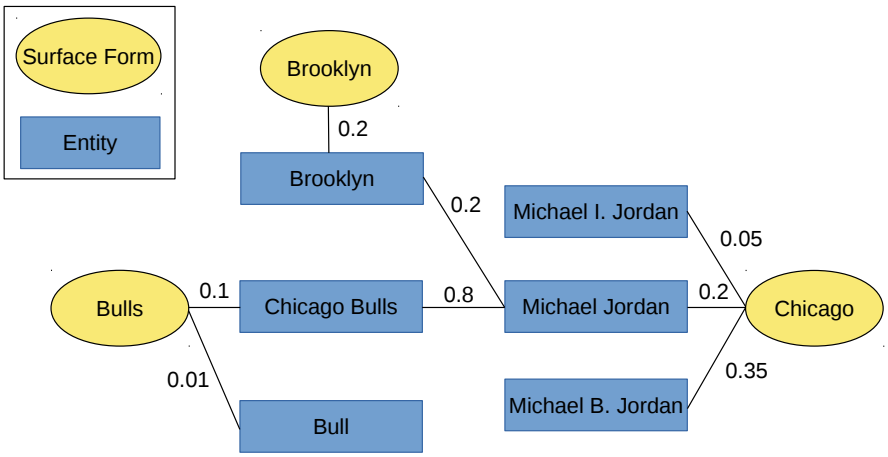



**Figure 3.9:** An example graph with surface forms, entities and transition weights

Graph algorithms typically do not use training data to weight the graph nodes (entities). Instead, training data in form of annotated entities is often used to compute graph features like transition probabilities (e.g., entity relatedness).

A simple yet effective and accurate approach for collectively linking Linked Data resources is *AGDISTIS* [Usb14], a KB-agnostic approach for RDF-KBs such as DBpedia or YAGO. All candidate entities across all surface forms form the set of initial nodes in a directed unweighted graph. The graph is extended with additional entities by performing a breadth-depth first approach starting at the candidate entities in the RDF graph. All entities within a given range are additionally added to the EL graph. Further, an edge between an entity pair is added if there is a direct relation between both entities in the KB. To rank all candidate entities, Usbeck et al. [Usb14] applied the Hyperlink-Induced Topic Search (HITS) algorithm [Kle99] and suggested to assign those target entities that provide the highest authority values.

Alhelbawy et al. [Alh14a; Alh14b] collectively linked entities with the help of an undirected weighted candidate entity graph. Depending on the entity relatedness measure, the edge weights are either binary or normalized similarity values between 0 and 1. First, the authors computed a local compatibility score $IConf(e_j)$ for each candidate entity in the graph (e.g., either Sense Prior or cosine similarity between surface form and name of candidate entity) to indicate how good candidate entities fit to a given surface form. Finally, three different approaches were proposed to select the most relevant candidate entities. The clique

approach recursively searches cliques in a graph with binary edges between entities. The candidates of the highest scoring clique represent the disambiguated entities. Cliques are scored according to the highest $IConf$ values in the clique. The disambiguated entities are combined to a new node in the graph, whereby the nodes (candidates) of the already linked surface forms are removed. In the second approach, the authors applied the PageRank algorithm [Bri98], which ranks the graph nodes according to their overall importance within the graph. The third approach describes a combination of the PageRank scores with the $IConf$ scores to incorporate local compatibility and global coherence.

The EL systems *AIDA* by Hoffart et al. [Hof11], *WAT* by Piccinno and Ferragina [Pic14] and the work by Han et al. [Han11b] also denote purely collective EL approaches. All methods rely on an undirected weighted graph where nodes represent surface forms AND candidate entities. The approaches integrate the textual context similarity of a candidate entity and its surface form (i.e., edge weight of a surface form - candidate entity pair) and the global interdependence between different EL decisions (i.e., edge weight of a candidate entity pair). Edges between surface form nodes and candidate entity nodes are only added if the entity is a candidate of the respective surface form. Moreover, edges between an entity pair are only added if the entity relatedness value between both entities is $> 0$. Hoffart et al. [Hof11] defined the weights for surface form - entity edges as a linear combination of the Sense Prior and a textual context matching score. In contrast, Han et al. [Han11b] simply used the cosine similarity between the textual surface form context and the entity description as edge weight. Piccinno and Ferragina [Pic14] computed context matching using the BM-25 similarity score [Jon00]. In order to compute an entity relatedness score, all works leverage the WLM [Mil08a]. To rank the candidate entities modeled within the graphs, Han et al. [Han11b] and Piccinno and Ferragina [Pic14] applied the (Personalized) PageRank algorithm [Hav03] or HITS algorithm [Kle99]. *AIDA* [Hof11], however, computes a dense subgraph ideally containing one surface form - entity edge for each surface form. With the problem being NP-hard, the authors approximated the problem with an extended greedy algorithm proposed in [Soz10]. This algorithm iteratively removes the entity nodes in the graph that have the smallest weighted degree (i.e., total weight of the incident edges).

The *Babelfy* system by Moro et al. [Mor14] is based on the semantic network *BabelNet* [Nav12] that contains Wikipedia entities and other concepts. Each node in the underlying directed graph represents a surface form AND the corresponding candidate in form of an entity or another concept. Further, the authors connected two candidate meanings of different surface forms if one is in the semantic signature of the other. A semantic signature of an entity can be seen as a set of highly related other entities. More details about the computation of semantic signatures can be found in the original work [Nav12]. Anyway, in order to drastically reduce the degree of ambiguity while keeping the coherence as high as possible, the authors proposed a novel densest subgraph heuristic. The resulting subgraph contains the most important nodes in terms of coherence. The remaining candidate entities are then ranked by its normalized weighted degree in this subgraph. The weight of a node is the fraction of other nodes in the graph to which the candidate entity is connected to.

Other graph-based approaches were proposed to collectively link entities in tweets. For instance, Shen et al. [She13] proposed the tweet linking system *KAURI*. One assumption

is that every twitter user has his own interest distribution over various named entities. However, the underlying disambiguation graph comprises all candidate entities of multiple surface forms across several tweets. Each node provides an intra-tweet local information score, which is computed with a LTR approach incorporating the Sense Prior, textual context similarity (cosine similarity of TF-IDF weighted context/entity description vectors), and coherence between entities within the same tweet (using WLM [Mil08a]). The graph edges denote the coherence between all candidate entities in the graph. Similar to the approaches described before, the authors used the WLM as entity relatedness measure and weight the graph edges accordingly. Finally, a personalized PageRank algorithm [Hav03] ranks the candidate entities across all surface forms by considering the intra-tweet local information score of each node and the respective user interest information.

Another collective, tweet Wikification approach by Huang et al. [Hua14] also creates an undirected, weighted disambiguation graph, where each node represents a surface form - candidate entity pair. Moreover, the approach is based on the following three principles:

- **Local compatibility:** Two pairs of a surface form - candidate entity combination $< e_j, m_i >$ that both provide a strong local compatibility between each other, tend to have similar characteristics. For instance, a surface form and a corresponding candidate entity usually share a set of characteristics like string similarity between $m_i$ and $e_k$ (e.g., the pairs $< H1N1, H1N1 >$ and $< Chicago, Chicago >$ have a high local compatibility in terms of similar labels). The local compatibility score between two pairs is defined as the cosine similarity between the respective local feature vectors.
- **Coreference:** If two surface forms like 'North Carolina' and 'nc' are coreferential, then both should be linked to the entity *North Carolina*. The coreference score of a surface form pair $m_i$ and $m_h$ is 1, if both provide the same label or one is an abbreviation of the other one. Otherwise the score is 0.
- **Semantic relatedness:** Two semantically related surface forms are more likely linked to entities that are also semantically related. The relatedness score between two nodes is defined as a combination of the relatedness between both surface forms and the relatedness between both candidate entities (computed with WLM [Mil08a]).

After computing the three features, they are combined via linear combination to calculate an edge weight between a node pair in the graph. To select the surface forms' target entities, the authors proposed a semi-supervised graph regularization framework based on the graph-based learning framework in [Zhu03]. More information can be found in [Hua14].

In **summary**, we state that graph-based algorithms are perfectly suited for collective EL. Further, these approaches have been very well researched across different domains (e.g., general domain, biomedical domain [Zhe14]). Basically, all graph-based approaches are somehow related and mainly differ in the employed feature set to compute the edge weights between candidate entities. To rank all graph nodes, the PageRank algorithm [Bri98] was applied in many works since the algorithm has been well researched in the last decade.

### 3.3.5 Probabilistic Approaches

Probabilistic EL approaches aim to find the most likely entity assignment given a surface form and its surrounding context. More formally, we compute

$$\underset{\Gamma}{\text{argmax}}\ p(\Gamma|M,C) = \underset{\Gamma}{\text{argmax}}\ \frac{p(\Gamma,M,C)}{p(M,C)} = \underset{\Gamma}{\text{argmax}}\ p(\Gamma,M,C) \tag{3.23}$$

i.e., the most likely configuration $\Gamma$ given the tuple of surface forms $M$ and the respective surrounding contexts $C$ [She14]. We note that we apply the basic notation of EL introduced in Section 2.1.1. Assuming that $M$ and $C$ are conditionally independent given $\Gamma$, the works [Gan16; Han11a; She14] obtain the following factorial expression for the joint model:

$$p(\Gamma,M,C) = p(\Gamma)\prod_{i=1}^{|M|} p(m_i|t_j^i)p(c_i|t_j^i) \tag{3.24}$$

Ganea et al. [Gan16] reformulated Equation 3.24 to the following *Probabilistic Bag-of-Hyperlinks* model, the current publicly available, state-of-the-art collective EL model:

$$log\ p(\Gamma|M,C) = \sum_{i=1}^{|M|}\left(log\ p(t_j^i|m_i) + \zeta \sum_{w_l \in c_i} log\ p(w_l|t_j^i)\right) + \tau \sum_{i<h} log\frac{p(t_j^i,t_k^h)}{p(t_j^i)p(t_k^h)} \tag{3.25}$$

with $\zeta$,$\tau$ denoting parameters to control the importance of the entity contexts and the entity-entity interactions. To estimate the probabilities in Equation 3.25, the authors made use of an underlying document-centric KB like Wikipedia. Based on this KB, they derived $p(t_j^i|m_i)$ by counting the entity occurrences given a surface form (i.e., Sense Prior). Moreover, they estimated $p(w_l|t_j^i)$ by counting how often word $w_l$ appears in the context window of an annotation of $t_j^i$ (cf. Equation 3.4 on Page 35). Probability $p(t_j^i,t_k^h)$ was estimated by counting the pairwise co-occurrences of the (assigned) entities $t_j^i$ and $t_k^h$ within the same Wikipedia document. To collectively link all surface forms in a document, Ganea et al. [Gan16] applied a loopy believe propagation [Mur99] technique that approximates the solution in polynomial time.

The authors of [Han11a; She14] reduced Equation 3.24 to the following non-collective form and, thus, linked all surface forms to each corresponding entity separately:

$$p(m_i,e_j,c_i) = p(e_j)p(m_i|e_j)p(c_i|e_j) \tag{3.26}$$

Typically, probability $p(e_j)$ denotes the Entity Prior and probability $p(m_i|e_j)$ denotes the Sense Prior. However, Shen et al. [She14] derived $p(e_j)$ with the PageRank algorithm on a given entity-centric KB and suggested to omit probability $p(m_i|e_j)$ due to the assumption of $p(m_i|e_j)$ being uniformly distributed. In terms of estimating the context probability, Han et al. [Han11a] relied on the unigram language model proposed in Section 3.2.4 on Page 34. In contrast, in [She14], the probability $p(c_i|e_j)$ refers to the proposed entity object

model, that captures the probability of words (or other objects) appearing near entity $e_j$. The probability of observing a specific object given entity $e_j$ is estimated from entity $e_j$'s network in a heterogeneous information network. More specifically, the authors applied *meta-path constrained random walks* [Lao10] to capture all probabilities $p(c_i|e_j)$.

In order to link surface forms in tables, Limaye et al. [Lim10] and Mulwad et al. [Mul13] suggested to collectively annotate table cells with entities, table columns with types, and pairs of table columns with relations. The authors modeled the interconnection of entities, types and relations with a number of random variables integrated in a graphical model [Kol09]. To collectively infer the optimal annotations, one has to maximize the joint probability of the random variables, which is NP-hard. For that purpose, the authors resorted to an approximation algorithm, specifically message-passing or belief propagation in factor graphs [Ksc01]. A very similar approach was leveraged by Kulkarni et al. [Kul09], who incorporated local features and pairwise topical coherence between candidate entities (global feature) in a graphical model to collectively annotate entities in Web documents. To solve the inference problem, the authors compared hill-climbing and linear program relaxations techniques. Another approach based on graphical models is integrated in *ZenCrowd* [Dem12], a large-scale EL framework using crowdsourcing techniques. The framework relies on human workers if the machine-based techniques do not provide enough confidence for an EL decision. More specifically, *ZenCrowd* generates micro-tasks which are published on a crowdsourcing platform. After a set of human workers performed these micro-tasks, the results are fed to back to the probabilistic reasoning framework, which generates the final result based on the respective annotations.

Very popular probabilistic methods for EL are topic models. We have already provided an overview of topic models and their architecture in our textual context feature Section 3.2.4. When utilizing topic models, the inference task involves the topic (entity) assignment for each surface form in a given document, i.e., $z_d = \{z_{m_1},...,z_{m_S}\}$ (relying on notation as introduced in Section 3.2.4 on Page 35). While Houlsby and Ciaramita [Hou14] proposed their own Gibbs sampler to effectively estimate $z_{m_i}$ for surface form $m_i$, most other works rely on an incremental Gibbs Sampler (e.g., [Han12; Kat11; Li16]). The incremental Gibbs Sampling algorithm rejuvenates old entity assignments in the light of new documents. More specifically, the topic model is trained on previously labeled documents $W$ first (e.g., Wikipedia). Given this model, the incremental Gibbs sampling algorithm is run on the set of documents $W \cup D$ and samples topics for surface forms in $D$ only, while keeping the topic assignments $z_k$ for the words $w_k$ in $W$ fixed [Kat11]. Then, in each incremental Gibbs sampling step during the annotation phase, only the assignments $z_{m_i}$ and $z_k$ in $D$ change, while those in $W$ remain constant. Finally, the topic (entity) assignments after a fixed number of steps represent the disambiguated entities.

In **summary**, we state that probabilistic EL frameworks were frequently employed to link entities. These methods allow to integrate local and global features equally. Further, sophisticated methods exist to resolve the optimization problem accurately and efficiently.

### 3.3.6 Classification Approaches

The task of EL can be also seen as a classification problem. With each entity $e_j$ representing a distinct class, a surface form $m_i$ has to be assigned to a specific class (entity). Classifiers

are typically supervised learning approaches and, thus, rely on a considerable amount of training data. Each training sample describes a surface form - candidate entity pair $(m_i, e_j)$ and the correct decision whether the candidate is relevant (e.g., 'positive') or not (e.g., 'negative'). During the training and classification step, each surface form - candidate entity pair is represented as a feature vector $\phi(m_i, e_j)$ that is composed with single features (cf. Section 3.2). The final classification accuracy then strongly depends on the amount of training data available.

Classification methods can be roughly divided into linear (e.g., Naive Bayes, Logistic Regression) and non-linear (e.g., $k$-Nearest-Neighbors) classification methods. For example, a linear two-class classification problem can be seen as splitting a high-dimensional input space with a hyperplane, with all points on a side being classified as 'positive' and all other points being classified as 'negative'. For an in-depth overview of classification methods, we refer to [Bis06]. EL approaches can be categorized into *binary* and *multi-class* classification techniques. Binary classifiers decide whether a candidate entity $e_j$ is a correct entity of surface form $m_i$ (in the following denoted as $e(m_i) = e_j$) by assigning a label $y$:

$$y(\phi(m_i,e_j)) = \begin{cases} +1, & \text{if } e(m_i) = e_j \\ -1, & \text{else.} \end{cases} \tag{3.27}$$

In multi-class classification approaches each candidate entity $e_j$ represents a distinctive class. Further, the classifier assigns exactly one class (entity) to a given surface form $m_i$.

When using a binary classification method, the classification process may indicate two or more candidate entities as 'positive' or relevant for a given surface form. For that reason, the results have to be ranked or further classified to pick the most relevant entity, depending on how EL is defined. If a single entity should be returned as disambiguation result (which is the case in most works), different methods are employed to select the most relevant one. These are, for instance, confidence-based methods [Pil11; Var09], VSM-based methods [Zha10a] or SVM-ranking models [Zha10b].

For the binary classifier, alongside with binary logistic regression methods [Mon11], most systems (e.g., [Pil11; Zha10a; Zha10b]) employ SVMs [Bos92]. The main idea of SVM's is to learn a hyperplane from the training data that separates the positive examples from the negative examples. Thereby, the learned hyperplane in the hyperspace maximizes the distance to the closest positive and negative training examples. In the biomedical domain, binary classification methods like SVMs, Decision Trees and Naive Bayes are often used to decide whether an entity denotes a gene or protein [Che06] (i.e., resolving gene-protein name ambiguity). In this special case, the entity has already been identified but the entity's class (gene or protein) is essential to assign the correct entity identifier.

Basically, SVM's can also be employed for multi-class classification by combining several binary classifiers in one-vs-one or one-vs-all fashion [Hsu02]. However, Pilz and Paaß [Pil11] argued that with increasing the number of entities in a KB, the classification problem rapidly becomes computationally more expensive and intractable. For that reason, Varma at al. [Var09] performed EL using a $k$-Nearest-Neighbors classifier. Another multi-class classification approach was proposed by Guo et al. [Guo13]. They applied Structured SVMs [Tso05] for EL in tweets that jointly optimize surface form detection (i.e., Entity

Recognition) and EL as a single end-to-end task. The system combines several context-sensitive and entity-entity relationship features and outperforms other state-of-the-art systems on tweet data sets.

In **summary**, we state that traditional classification approaches can also be applied to link entities. However, recently proposed state-of-the-art approaches do not rely on classification methods due to a difficult integration of topical coherence features for collective EL. For that reason, most classification EL approaches are entity-independent ranking methods. However, classification EL methods achieved (nearly) state-of-the-art results in the TAC-KBP task (cf. Section 2.1.2).

### 3.3.7 Model Combinations and Other Approaches

Model combinations, also known as ensemble methods, combine (different) learning algorithms with various characteristics and try to improve predictive performance [Ade05; Opi99]. In the context of EL, model combinations seek to overcome existing weaknesses of single variants.

For instance, Zhang et al. [Zha10b] were the first authors who combined different EL approaches. First, the following three separate stand-alone systems were constructed: (i) An Information-Retrieval-based system (cf. Section 3.3.2), (ii) a LTR-based system (cf. Section 3.3.3), and (iii) a binary classification system (cf. Section 3.3.6). All three approaches rely on standard features such as word-category pairs or string similarity measures. Finally, a three-class SVM classifier was trained to judge which of the three systems should be trusted. Ji and Grishman [Ji11a] also applied a voting approach on the best nine EL systems proposed in the context of the TAC-KBP2010 track and found that all system combinations achieved significant improvements, with the highest absolute improvement of 4.7 F1 percentage points overall. Another example of model combination was presented in [Che11], where the authors utilized composite functions like majority voting and weighted average to incorporate four supervised and four unsupervised baseline EL approaches. The results showed that a model combination achieves an accuracy gain of 1.3 F1 percentage points with the majority vote function and 0.5 F1 percentage points with the weighted average function over the best single variant. Furthermore, the *CUNY-UIUC-SRI* system [Cas11] combines the collaborative entity ranking framework by Chen and Ji [Che11] with *Glow*, the EL framework by Ratinov et al. [Rat11]. This combination led to an improvement of $\approx 2-3$ F1 percentage points compared to the baseline systems on the TAC-KBP2011 data set.

In the following, we briefly present two important works that do not fit in our classification scheme above and significantly contribute to the current state-of-the-art of EL. In 2014, the authors Guo et al. [Guo14] suggested to use a probability distribution based on a random walk with restart [Ton06]. The latter is employed on a subgraph of the input KB to represent the semantics of entities $e_j$ and input document $d$ (similar to the semantic signatures presented by Navigli et al. [Mor14]). The presented iterative algorithm links surface forms according to their degree of ambiguity. For a surface form $m_i$, the candidate entity $e_j$ that provides the highest semantic similarity (Kullback-Leibler divergence [Kul51] of the probability distributions) to document $d$ is selected as correct entity. After each iteration the document's semantic signature is updated accordingly.

Another important approach was proposed by Cheng and Roth [Che13], which is integrated in the publicly available state-of-the-art EL system *Wikifier*. The authors formulated their collective approach as an Integer Linear Program (ILP). Overall, they used two boolean variables: Variable $s_j^i$ denotes whether we assign surface form $m_i$ to the target entity $t_j^i$. Variable $r_{jk}^{(i,h)}$ denotes that the entity assignments $t_j^i$ and $t_k^h$ are made simultaneously, that is, $r_{jk}^{(i,h)} = t_j^i \wedge t_k^h$. Moreover, variable $p_j^i$ denotes the Sense Prior of the assigned entity $t_j^i$ of surface form $m_i$. Finally, variable $w_{jk}^{(i,h)}$ denotes the confidence of finding a relation between the entity assignments $t_j^i$ and $t_k^h$. Since relation extraction and the underlying score computation is out of scope in this work, we refer the interested reader to the main work [Che13]. However, the authors found the best entity assignment $\Gamma$ by solving the following ILP problem:

$$\Gamma = \underset{\Gamma}{\operatorname{argmax}} \sum_i \sum_j p_j^i s_j^i + \sum_{i,h} \sum_{j,k} w_{jk}^{(i,h)} r_{jk}^{(i,h)} \tag{3.28}$$

$$\begin{aligned}
\text{s.t.} \quad & r_{jk}^{(i,h)} \in \{0,1\} && \text{Integral constraints} \\
& s_j^i \in \{0,1\} && \text{Integral constraints} \\
& \forall i \sum_j s_j^i = 1 && \text{Unique solution} \\
& 2r_{jk}^{(i,h)} \leq s_j^i + s_k^h && \text{Relation definition}
\end{aligned}$$

Since $w_{jk}^{(i,h)} = 0$ for most entity pairs is considered, the resulting ILP is tractable and can be solved by standard ILP solvers.

## 3.4 Abstaining

So far, we presented how candidate entities are generated and how these are ranked to determine the correct target entity. In practice, in a document $d$, there might be a list of surface forms $S_{NIL}$ whose correct target entities are not defined in the underlying KB (i.e., $e^{m_i} = NIL$). Therefore, EL systems have to deal with the problem of predicting not linkable surface forms and, thus, abstain by linking those surface forms to the pseudo-entity *NIL*. For simplification, $NIL$ is typically considered as an additional entity in the KB (cf. problem formulation in Section 2.1.1). Many works assume that the correct target entity is constantly available in the KB and ignore not linkable surface form prediction. However, we provide a brief overview of the main abstaining methods in the following.

Typically, EL algorithms return $NIL$ in the following situations:

- If no candidate entities are found for a surface form during the candidate entity generation step.
- If the algorithm is uncertain about the correct entity assignment during the disambiguation step.

In terms of candidate generation, a simple heuristic is to annotate $NIL$ if the set of candidate entities for a surface form is empty (e.g., in [Che10; Nem10; Var09]). This is a

reliable procedure if the underlying surface form - entity dictionary is kept small and/or an exact matching technique is applied. If a fuzzy matching technique is employed to circumvent spelling mistakes in surface forms, the number of candidate entities typically increases significantly. Hence, the number of correct $NIL$ annotations is reduced.

Besides the method during candidate generation, many approaches integrate an abstaining mechanism in the main ranking step. For instance, some approaches integrate a $NIL$ threshold to predict a not linkable surface form (e.g., [Bun06; Kul09; Li13]). More specifically, in these approaches the top-ranked candidate entity $e_{top}$ is associated with a score $s_{top}$. If $s_{top}$ is smaller than a given $NIL$ threshold the surface form is annotated with $NIL$. The respective threshold is automatically learned from the training data. Another approach is to use supervised machine learning techniques. For instance, the works [Rat11; Zha11a; Zhe10] train a binary classifier that predicts whether the surface form - top candidate entity pair $< m_i, e_{top} >$ is the correct mapping. Here, a $< m_i, e_{top} >$ pair denotes a feature vector mainly comprising features presented in Section 3.2. Ratinov et al. [Rat11] and Zheng et al. [Zhe10] additionally incorporated the feature whether the surface form is detected as a named entity by a Named Entity Recognition system. All mentioned systems employ a SVM for classification.

Other systems directly integrate the abstaining mechanism into the disambiguation process. For instance, Dredze et al. [Dre10] used the LTR framework and assumed $NIL$ to be a distinct candidate. If the LTR framework ranks $NIL$ at the top, the surface form is considered as not linkable. Otherwise, the top-ranked entity is returned as the mapping entity. Another example is the probabilistic approach by Han and Sun [Han11a]. The authors assumed that for a surface form that refers to a specific candidate entity, the probability of generating this surface form by the candidate entity's language model should by significantly higher than the probability of this surface form being generated by a general language model. Basically, the EL model adds a $NIL$ entity to the underlying KB and assumes that $NIL$ generates a surface form according to the general language model. If the probability of the surface form being generated by $NIL$ is higher than the probability of each candidate entity generating the surface form, then the surface form is considered as not linkable [Han11a; She15].

To **summarize**, abstaining is a very important task in EL algorithms when it comes to linking surface forms whose referent entity is not in the underlying KB. Different approaches were proposed to tackle this problem, but, depending on the evaluated data sets, the results were not always convincing. Additional work must be invested to further improve the abstaining results.

## 3.5 Conclusion

In this chapter, we provided an in-depth overview of existing (state-of-the-art) EL approaches proposed in the literature. More specifically, we distinguished between different yet crucial components that are necessary for accurate EL. These are (i) candidate entity generation, (ii) EL features, (iii) disambiguation, and (iv) abstaining. Candidate generation is important in terms of reducing computational complexity and improving EL accuracy. In this step, we select a (small) set of relevant candidate entities for each surface form. We distinguished between name dictionary methods, surface form expansion methods

and search engine methods. In the EL features section, we reviewed the most relevant features found to be useful and important in terms of ranking candidate entities accurately. Here, we distinguished between context-independent (i.e., entity name, entity popularity and entity type features) and context-dependent entity features (i.e., textual context and topical coherence features). In our third main section, we subdivided entity disambiguation algorithms into VSM approaches, Information Retrieval approaches, LTR approaches, graph-based approaches, probabilistic approaches, classification approaches and ensemble approaches. Finally, in the abstaining section, we briefly reported methods that are used to abstain if no appropriate candidate entity is available in the KB.

Overall, we presented and explained a wide range of different methods but focused on the most important techniques in each section. We did not categorize general-domain and special-domain EL since special-domain approaches typically rely on existing methods with a stronger focus on specific feature selection (i.e., special-domain features, for instance, gene length in the gene domain). We have not found exhaustive surveys in the literature that cover all or multiple domains, or provide a broad view on the topic. Instead, we suggest the surveys for general-domain EL [Lin15; She15] and our survey about EL in the biomedical domain [Zwi15b] for further reading.

# Part III

## Robust Entity Linking

# CHAPTER 4

## Robustness in Entity Linking Systems

This chapter introduces the main part of this work and provides an extensive overview and summary of the following chapters. More specifically, we derive our research questions based on the limitations of existing Entity Linking (EL) systems in the literature (cf. Chapter 3) and provide an overview of our contributions made in this work. Figure 4.1 shows a structural overview of this chapter. First, we identify and discuss three core limitations concerning (state-of-the-art) EL approaches in Section 4.1. Based on these limitations we introduce and define the term *Robustness* in the context of EL systems and pose our main research question of this work in Section 4.2. In Section 4.3, we analyze the typical structure of an EL algorithm. We identify three crucial components, i.e., (i) the underlying knowledge base (KB), (ii) the entity relatedness measure, and (iii) the textual context matching technique, which (significantly) contribute to robust EL. In the Sections 4.4, 4.5 and 4.6, we pose research questions for each of the identified components and summarize our contributions made in the respective chapters. Finally, in Section 4.7, we briefly describe our main contribution, *DoSeR*, a robust EL system that combines the findings and outcomes of this work.

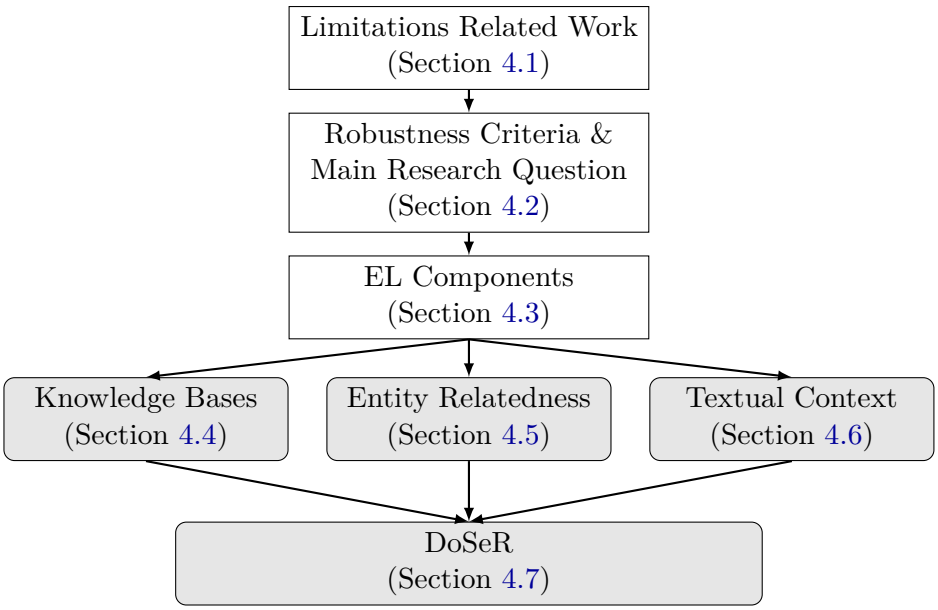



**Figure 4.1:** Overview of this chapter. Grey boxes denote the contributions in our work.

## 4.1 Limitations in Related Work

The EL task has been extensively studied over recent years. Different communities, such as the Natural Language Processing, Semantic Web, Data Mining and Biomedical community, have brought up a plethora of algorithms and techniques to tackle the problem of linking surface forms to entities within a KB. An in-depth overview of these methods can be found in Chapter 2 and Chapter 3 in this work. In the following, we identify three core limitations concerning (state-of-the-art) EL approaches.

***Domain Dependency***

Researchers from various communities addressed the problems of EL and focused on linking general-domain entities like cities, persons, organizations, etc. [She12b; Usb14]. Moreover, several subcommunities have evolved that particularly focus on linking special-domain entities (e.g., biomedical domain [Zwi15b], earth science domain [Wan15]). In the biomedical domain, for instance, the focus lies on disambiguating genes and proteins [Tap05], or species [Har12]. When taking domain-specific entities, EL is more difficult due to special domain characteristics [Zwi15a]. For instance, special-domain KBs that contain biomedical entities often lack appropriate entity descriptions (e.g., genes) or exclusively comprise domain-related entities (e.g., UniProt focuses on genes only). Unfortunately, EL systems that have been developed to work on a specific domain have barely been evaluated on other domains [Cam16]. Thus, little knowledge is available for how EL approaches from one domain perform on other domains. Recently, a first attempt to analyze EL accuracy of well-known EL systems across different domains was made by Thorne et al. [Cam16]. The authors analyzed how the well-known EL frameworks *TagMe* [Fer12] and *Babelfy* [Mor14] perform in linking biomedical entities. They showed inferior results in comparison to *MetaMap*, a tool for recognizing and linking entities in the UMLS KB. So far, if surface forms from various domains should be disambiguated, different EL approaches are commonly employed since robust domain-agnostic approaches are missing. It is still an open question how to achieve state-of-the-art results on multiple domains with a single EL system.

***Knowledge Base Properties***

Apart from the lack of domain-agnostic approaches, a related problem occurs when EL systems consider entities from multiple domains or KBs, respectively. For instance, several general-domain KBs like DBpedia or Wikipedia are well-known for their wide-ranging and high quality entities. These KBs also comprise a broad range of popular entities from several specific domains (e.g., *Influenza*) but lack very specific entities [Tia13] (e.g., *IIV3-011L* gene). If EL systems should cover a broader range of entities than given by single KB, multiple KBs have to be considered. A combination of multiple KBs may lead to (extremely) large and heterogeneous KBs. We assume that adding additional entities to KBs results in decreasing EL results and performance since more entities can be considered to be candidate entities for surface forms.

In the context of differently scaled KBs, a related problem has emerged when comparing non-publicly available EL approaches. Re-implementing the respective algorithms is not an absolutely fair method to compare the approaches: Usually crucial implementation details remain unknown in the original publications. As a consequence, many EL systems were

compared on the same data sets via direct result comparison based on the assumption that the versions of the underlying KBs are exactly the same. Unfortunately, this is not the case in praxis. For instance, when using Wikipedia as KB, the authors have to assure that both EL systems use the exact same version of Wikipedia to provide an absolutely fair comparison since newer versions contain more entities and/or modified entity descriptions. Overall, it is still an open question how EL approaches perform with differently sized and/or multiple (heterogeneous) KBs.

EL systems that achieve state-of-the-art results on one or more data sets are often optimized on a specific KB to perfectly leverage the underlying entity definitions. However, different KBs exist that contain the same kind of entities but differ in the way the entities are described. Two popular examples are the document-centric KBs Wikipedia and CalbC as well as the corresponding entity-centric KBs DBpedia and UMLS/Uniprot etc. While many approaches partially extract entity information from different KBs (e.g., surface forms in document-centric KBs), most EL approaches are strongly adapted to specific KBs. Recently, some researchers recognized this deficiency and started to construct EL systems that are KB-agnostic in terms of a specific KB type. For instance, in 2014, Usbeck et al. [Usb14] proposed *AGDISTIS*, a KB-agnostic framework in terms of RDF-KBs to disambiguate Linked Data resources. Despite representing a simple approach, *AGDISTIS* achieves state-of-the-art results on some data sets. The more sophisticated, probabilistic approach *SHINE* by Shen et al. [She14] links surface forms in web texts to named entities in an arbitrary Heterogeneous Information Networks. Although the authors achieved strong EL results, they did not evaluate their approach on KBs other than DBLP. Thus, a comparison with state-of-the-art approaches for Wikipedia/DBpedia is missing.

Depending on the domain and popularity of entities, different amounts of training data are available to optimize the EL system. While Wikipedia provides a huge number of manually annotated entities via interlinks between Wikipedia articles, other more specific domains often lack such valuable entity information. The respective approaches have to get along with little or no training data. Little information is available on how current state-of-the-art techniques employed in EL systems perform with a reduced amount of training data or limited entity information in KBs. In addition, depending on how KB information was created and curated, entity information might be erroneous. This is the case, if entity annotations within a corpus were automatically created with an Entity Recognition and another EL system. It is still unknown, how sensitive EL algorithms are regarding noisy training data/KB information.

***Document Structures and Types***

Authors that propose new EL approaches commonly focus on linking surface forms within specific document structures and types. For instance, Usbeck et al. [Usb14] linked entities in web documents, while Limaye et al. [Usb14] and Huang et al. [Hua14] focused on tables and Twitter tweets, respectively. Generally, table and tweet EL systems are adapted to the respective document structure because of limited or missing (textual) context information. In these approaches, other information, such as column types and relationships between columns in tables or related tweet information by the same user, is exploited instead. However, these perfectly optimized approaches cannot be applied to general documents

without (significant) limitations in terms of EL accuracy. Apart from various EL approaches for different document structures (e.g., tables) and types (e.g., news documents), many state-of-the-art EL algorithms were evaluated on a specific data set. A very popular data set that has often been used to optimize and/or evaluate the constructed annotation system is AIDA [Hof11], originally derived from the Co-NLL 2003 shared task. This data set contains a lot of documents with similar tables and short introductory textual descriptions. Anyway, despite this characteristic document structure, the underlying algorithms were often exclusively evaluated on this data set (e.g. [Alh14b; Hof11]) or in combination with few (i.e., one or two) others (e.g. [Bar14; Hua15]). Unfortunately, most state-of-the-art EL systems are not publicly available and, hence, cannot be evaluated on other data sets to analyze the robustness in terms of different document structures and types.

In **summary**, we identified three core limitations in terms of existing EL systems, namely domain dependency, knowledge base properties as well as document structures and types:

- **Domain dependency** describes the lack of domain-agnostic, state-of-the-art EL systems (e.g., general-domain, biomedical domain, earth science domain, etc.).
- **Knowledge base properties** such as the quantity and quality of entity definitions and training data may significantly influence EL results and are omni-present when dealing with unpopular or special-domain entities. This raises questions on how EL systems perform with large-scale and heterogeneous KBs and a poor quality and low quantity of entity definitions. In this context, most EL systems are not able to leverage the knowledge of differently structured KBs to mitigate negative EL results.
- **Document structures and types** emphasizes that state-of-the-art EL systems have been (strongly) optimized for specific document structures (e.g., tables) and types (e.g., news articles).

Overall, we note that many researchers optimized their EL approach for a specific domain/type of KB and for specific document structures and types. This led to very specialized systems that are only applicable for very specific tasks. Based on these limitations, we pose our main research question in Section 4.2.

## 4.2 Main Research Question: Robustness in Entity Linking Systems

After addressing several core limitations of existing EL approaches, we now introduce the term ***Robustness*** in the context of EL systems. We define Robustness as an umbrella term that covers two crucial characteristics of EL systems, namely Structural Robustness and Consistency. Structural Robustness describes the ability of EL systems to leverage structurally different data (e.g., entity-centric and document-centric KBs). Consistency of EL systems, however, describes consistent results with different data properties (e.g., differently sized KBs). More specifically, we define the characteristics as follows:

- **Structural Robustness:** Basically, a structurally robust EL system has the ability to utilize structurally different KBs, i.e., entity-centric and document-centric KBs. A combination of both types can lead to a complementation in terms of entity coverage, i.e., the total number of entities available in a KB, and entity definitions,

i.e., the completeness and quality of the description of one entity. Moreover, Structural Robustness also defines achieving high EL accuracy over different document structures (e.g., tables) and types (e.g., news articles).

- **Consistency:** Depending on the underlying domain/KB, different amounts of entity definitions and training data in the form of manually annotated entities in a corpus are available. In specific domains, the quantity and quality of available annotated documents is generally (very) limited. Additionally, popular entities typically provide high-quality entity definitions while unpopular entities often lack necessary, EL relevant information and training data. Hence, algorithms should perform sufficiently well without extensive training data as in the case of unpopular entities or special-domain entities. Moreover, EL systems should cope with large-scale and heterogeneous KBs without accuracy loss to be able to cover a broad range of KBs. Depending on how KBs were created and curated, entity definitions may contain errors in the form of wrong or missing relations in entity-centric KBs or wrong annotations in document-centric KBs. EL systems should maintain a high linking quality if entity definitions or training data is noisy to a certain extent. Moreover, an algorithm provides Consistency when it achieves convincing results on different domains, for instance on general-domain KBs like Wikipedia and specific-domain KBs like DBLP (i.e., computer science domain) or CalbC (i.e., biomedical domain).

Table 4.1 contrasts the characteristics Structural Robustness and Consistency and summarizes the respective criteria.

**Table 4.1:** Overview of Structural Robustness and Consistency criteria

| **Robustness** | |
|---|---|
| **Structural Robustness** | **Consistency** |
| Applicable on (with state-of-the-art results): | Consistent results with/on: |
| - Different types of KBs | - Various domains |
| - Different document structures and types | - Large-scale and heterogeneous KBs |
| | - Low quantity and poor quality of entity data |

Robustness in EL systems is still an open problem and it is required to avoid a plethora of stand-alone systems that are highly optimized and only applicable for specific domains, KBs and/or data sets. Hence, our main research question in this work is as follows:

**Main Research Question:** *Which EL system achieves state-of-the-art results while providing Structural Robustness and Consistency?*

In order to create such a system, we start by identifying and investigating important components of EL systems separately and analyze how these affect different robustness criteria. In the following Section 4.3, we first analyze the components of EL systems.

## 4.3 Components of Entity Linking Systems

The main goal of this work is to create a robust EL system based on previously evaluated components in terms of robustness criteria. For this purpose, we first analyze typical components of an EL system in this section. Despite the literature offering a huge variety of different EL algorithms and techniques, the structures of EL algorithms resemble each other. Figure 4.2 shows an overview of those components that are basically used to link surface forms to entities. Each box represents an important step in EL systems. Depending on the system itself, some components are not mandatory and can be omitted.

The first crucial step (Step (i) in Figure 4.2) is mandatory for all EL systems and requires to select an underlying KB that comprises the set of target entities. As described in Section 2.2, we distinguish between entity-centric and document-centric KBs. Each KB type offers different kinds of entity definitions and various ways to leverage the underlying information. In this context, it is important to note that many authors typically construct a system-adapted EL index exclusively containing those entity information that are used by their respective algorithms. This provides several advantages including size-reduced KBs and faster information access. However, the basic KB type typically remains the same.

Based on the underlying KB, respective entity describing features are used to link surface forms to entities. A significant number of works generate a set of candidate entities for each surface form to improve accuracy and performance in the steps later on (Step (ii)). If the candidate entity generation step is omitted, the set of relevant entities for a surface form comprises all entities within the KB.

Next, re-capturing our feature classifications in the related work Section 3.2, we distinguish between context-independent (Step (iii)) and context-dependent features (Step (iv)). State-of-the-art EL algorithms typically leverage both types of features but may also perform well while only using one specific feature type. It is important to note that context-dependent features can be further classified into topical coherence and textual context features. Topical coherence features are used in collective EL approaches. In Figure 4.2, the light blue boxes represent respective feature examples. However, an in-depth feature overview can be found in Section 3.2 on Page 30.

Finally, after computing a specific feature set, the disambiguation algorithm combines the features and ranks the relevant candidate entities according to its feature scores (Step (v)). Disambiguation algorithms can be classified into different approaches. However, to improve clarity we omit subclassifications and refer to the respective Section 3.3 instead.

The shaded components *Knowledge Bases*, *Entity Relatedness* and *Textual Context* are further investigated in this work. We assume that these parts strongly contribute to Structural Robustness and Consistency. For instance, the KB is particularly important since all other EL components rely and depend on the (entity) data located in the KB. Furthermore, entity relatedness and textual context matching techniques significantly contribute to achieving state-of-the-art results. However, the effectiveness of these techniques strongly depends on external factors, such as the length of surface form context for context matching or the number of entity annotated documents for entity relatedness computation. Thus, these techniques are prone to perform poorly on specific document structures/types or without specific entity data.

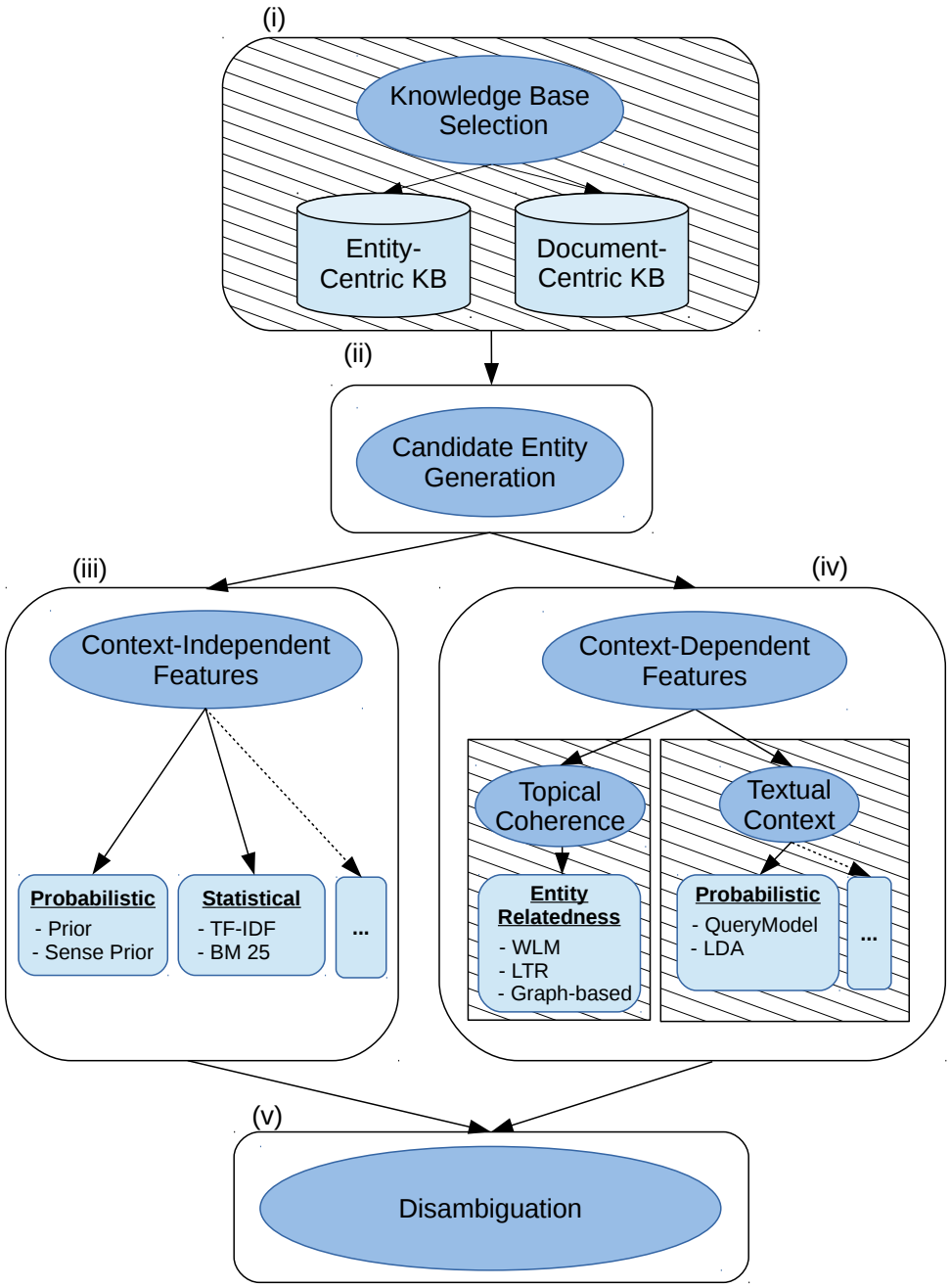



**Figure 4.2:** The general architecture of an EL system consists of five essential parts: (i) Knowledge base selection and preparation, (ii) optional candidate entity generation and selection, (iii) surface form matching, (iv) context matching, and (v) the main EL step where surface form features and/or context features are used or combined to create an overall result. The shaded areas play a crucial role in terms of robust EL and are further investigated in the context of this work.

## 4.4 Research Question I: Knowledge Bases

**Problem Setting:** Selecting a KB is the first crucial step when constructing an EL system. By holding relevant information for each entity, the underlying KB represents the fundamental frame for each system (cf. Figure 4.2). Basically, KBs play a particularly important part in robust EL since Structural Robustness and Consistency are tied to KBs. Structural Robustness in terms of KBs refers to EL systems being KB-agnostic in terms of the KBs' structures. In other words, the EL system is able to exploit different kinds of entity definitions as they occur for instance in entity-centric and document-centric KBs. This is mainly a task of feature and algorithm engineering. In contrast, Consistency directly refers to content-related KB properties. More specifically, a robust EL system should provide Consistency in terms of various domains, large-scale and heterogeneous KBs, and low quantity and poor quality of entity data (cf. Section 4.2). Although a plethora of works has been presented in the context of EL, it still remains unclear how and to which degree content-related KB properties influence EL results. Thus, we pose the following research question:

> **Research Question:** *How and to which extent do content-related KB properties influence EL results?*

**Contribution:** In Chapter 5, we extensively investigate how content-related KB properties influence EL results. These are (i) the entity format, i.e., intensional or extensional entity descriptions, (ii) user data, i.e., the quantity and quality of externally disambiguated entities, and (iii) the quantity and heterogeneity of entities to disambiguate, i.e., the number and size of different domains in a KB. To this end, we implemented three ranking-based EL algorithms for different entity formats (i.e., algorithms for entity-centric and document-centric KBs and a combination of both). Given the approaches, we investigate how EL results evolve with different degrees of quantity and quality of user data as well as with large-scale and heterogeneous KBs. In our experiments, we mainly rely on special-domain KBs (i.e., biomedical domain KBs) since limited user data and quantity and heterogeneity of entities to disambiguate are well-known issues in this domain [Zwi15a]. Our results show that (i) the choice of the entity format that is used to attain the best EL results strongly depends on the amount of available user data, (ii) the entity format strongly affects EL results with large-scale and heterogeneous KBs, (iii) EL results with all approaches are robust against a moderate amount of noise in user data, and (iv) a federated approach that combines the knowledge of intensional and extensional entity definitions can significantly increase the Consistency of (specialized) EL systems.

## 4.5 Research Question II: Entity Relatedness

**Problem Setting:** Context-dependent features represent a particularly important aspect in EL algorithms since the surrounding contexts of surface forms provide the necessary features to determine the correct target entity (cf. Chapter 3). One such context-dependent feature is entity coherence, which captures the coherence between entities of different surface forms within the same document. Typically, to compute the entity coherence within

a document, a (pairwise) entity relatedness measure is applied to compute a relatedness score between candidate entity pairs. Although the literature provides a huge number of different entity relatedness measures, nearly all techniques are tied to a specific KB or KB type [Usb14]. Further, it remains unclear how existing entity relatedness measures perform with a low quantity and poor quality of entity descriptions. For instance, wrong relations in entity-centric KBs or wrong (user) annotations in document-centric KBs might significantly decrease EL results. To achieve Structural Robustness and Consistency in an EL system, we need a KB-agnostic and accurate entity relatedness measure that is ideally robust against a low quantity and poor quality of entity definitions. Overall, we pose the following research question:

> **Research Question:** *Which entity relatedness measure provides Structural Robustness and Consistency while achieving state-of-the-art results in EL systems?*

**Contribution:** In Chapter 6, we propose a new state-of-the-art entity relatedness measure based on a neural network language model, namely Word2Vec. More specifically, we propose two algorithms that exploit the structure of entity-centric and document-centric KBs and generate appropriate input corpora for Word2Vec. In our evaluation, instead of re-implementing and directly comparing all current existing state-of-the-art measures on different data sets, we chose to integrate the new entity relatedness measure in a graph-based, collective baseline algorithm. Our simple algorithm (significantly) outperforms most existing, publicly available EL approaches. Further, we show that our measure is robust against a low quantity of underlying entity annotations as well as a moderate amount of noise in the underlying training data. In summary, we present a new state-of-the-art entity relatedness measure that provides Structural Robustness and Consistency.

## 4.6 Research Question III: Textual Context

**Problem Setting:** The textual surrounding context is another important context-dependent feature to link entities accurately. The surrounding context is typically made up of words and phrases before and after a surface form. Given a specific context matching technique, its accuracy in EL algorithms strongly depends on two main factors: (i) the textual surface form context, and (ii) the quality and quantity of the entity definitions located in a KB. While a tremendous number of various EL context matching techniques has been proposed (cf. Section 3.2.4), most techniques were evaluated on general-domain KBs with Wikipedia leading the way (e.g. [Gan16; Hou14; Sen12]. Wikipedia provides high quality and extensive entity descriptions that can be leveraged for highly accurate EL results. (Domain-specific,) entity-centric KBs often contain very short entity descriptions that lack important information (e.g., DBpedia, Uniprot). Moreover, in document-centric KBs, entities are described extensionally, i.e., through instances and usage [Ogd23], which requires the matching between a surface form context with other surface form contexts of the respective candidate entity. Unfortunately, it is unclear how the bulk of state-of-the-art context matching techniques cope with short surface form contexts (as is the case in tables and tweets) and/or short entity descriptions due to missing experiments in the literature.

To achieve Structural Robustness and Consistency in EL systems, we need a textual context matching technique that performs well on different kinds of KBs with variable length entity descriptions. Moreover, it should cope with long and short textual surface form contexts. Overall, we pose the following research question:

> **Research Question:** *Which context matching technique provides Structural Robustness and Consistency while achieving state-of-the-art results in EL systems?*

**Contribution:** In Chapter 7, we present the neural-network-based approach Doc2Vec as a robust textual context matching technique for EL systems. Doc2Vec is based on Word2Vec and allows us to create semantic embeddings of sentences, paragraphs and documents. We leverage entity descriptions located in entity-centric and document-centric KBs to construct these document embeddings. Further, we compare Doc2Vec to two TF-IDF-based approaches (i.e., Vector Space Model with TF-IDF weighted vectors and Okapi BM-25), a language model approach and an LDA approach. We show that Doc2Vec is robust against short surface form contexts as occurring in tables and tweets, variable length entity descriptions and different KB types. Moreover, we show that the Vector Space Model approach outperforms all other approaches if a sufficient amount of contextual and entity describing information is available. In summary, we suggest Doc2Vec as textual context matching approach for robust EL.

## 4.7 Main Contribution: DoSeR - A Robust Entity Linking Framework

Most existing EL systems are highly optimized toward a specific data set, KB or domain, but do not (fully) provide Structural Robustness and Consistency. To create such a robust EL system, we first analyze three crucial components of EL algorithms to gain new insights into techniques and algorithms whose usage essentially influence Robustness in Chapter 5, 6 and 7. In these chapters, we reveal the following three core findings in terms of robust EL, which are considered in our robust EL framework:

1. We show that a federated approach leveraging knowledge from entity-centric and document-centric KBs can (significantly) improve the Consistency of EL systems.
2. We present a new state-of-the-art entity relatedness measure for topical coherence computation that provides Structural Robustness and Consistency.
3. We present Doc2Vec as textual context matching technique that provides Structural Robustness and Consistency in terms of low quantity (short) entity descriptions.

Based on these findings and outcomes, we aim to construct a robust, state-of-the-art EL framework. More specifically, in Chapter 8, we present *DoSeR* (**D**isambiguation **o**f **Se**mantic **R**esources). DoSeR is a KB-agnostic EL framework that extracts relevant entity information from multiple (entity-centric and document-centric) KBs in a fully automatic way. Further, it creates indexes and models that are required by the used algorithms later on. DoSeR accepts different types of input documents such as tables, news articles and tweets whereby each document provides one or multiple, previously annotated surface forms. Our

main EL algorithm in DoSeR utilizes semantic entity and document embeddings for entity relatedness and textual context matching computation and represents a new collective, graph-based approach. The DoSeR algorithm is also able to abstain if no appropriate candidate entity can be found for a specific surface form. To evaluate the EL accuracy, we conducted experiments on general-domain KBs (e.g., Wikipedia, DBpedia, YAGO3) and special-domain KBs (e.g., Uniprot). In our evaluation, we compare DoSeR to other publicly (e.g., Wikifier [Rat11], AIDA [Hof11] and AGDISTIS [Usb14]) and non-publicly (e.g., Probabilistic Bag-Of-Hyperlinks model [Gan16]) available EL systems and discuss the achieved results in detail. In our experiments, DoSeR outperforms current state-of-the-art EL systems over a wide range of very different data sets and domains. Moreover, DoSeR provides Structural Robustness and Consistency in terms of most criteria. We also provide DoSeR as well as the underlying KBs as open source solutions.

Table 4.2 provides an overview of the conducted experiments in the respective chapters. A 'check' in parentheses indicates that this experiment was either not fully conducted and/or the outcomes are deduced from other experiments (some additional experiments may be required to fully confirm the results).

**Table 4.2:** Overview of conducted experiments in the respective chapters

| **Experiments** | **Knowledge Bases** Chapter 5 | **Entity Relatedness** Chapter 6 | **Textual Context** Chapter 7 | **DoSeR** Chapter 8 |
|---|---|---|---|---|
| Different Types of KBs | ✓ | ✓ | ✓ | ✓ |
| Different Document Structures | ✗ | ✓ | ✓ | ✓ |
| Various domains | (✓) | (✓) | ✗ | ✓ |
| Large and heterogeneous KBs | ✓ | ✗ | ✗ | ✗ |
| Low quantity of entity data | ✓ | (✓) | ✓ | ✓ |
| Poor quality of entity data | ✓ | ✓ | ✗ | (✓) |


DoSeR

# CHAPTER 5

## Knowledge Bases

In this chapter, we investigate how and to which extent various knowledge base (KB) properties influence Entity Linking (EL) results. The evaluated KB properties are (i) the entity format, i.e., the way entities are described (intensionally or extensionally), (ii) user data, i.e., the quantity and quality of externally disambiguated entities, and (iii) the quantity and heterogeneity of entities, i.e., the number and size of different domains in a KB. To this end, we implemented three ranking-based EL systems to address various entity definitions and provide a systematic evaluation of the defined KB properties in the biomedical domain. In our evaluation, we show that (i) the choice of the entity format to achieve the best EL results depends on the amount of available user data, (ii) the entity format strongly affects EL results with large-scale and heterogeneous KBs, (iii) all evaluated approaches are robust against a moderate amount of noise in user data, and (iv) a federated approach that leverages both entity formats (i.e., intensional and extensional entity definitions) can significantly improve the Consistency of EL systems. This chapter covers and combines the ideas, findings and materials published in the works [Zwi13b], [Zwi15a] and [Zwi15c].

The remainder of the chapter is structured as follows: In Section 5.1, we briefly introduce the chapter's core question, the contributions and the results. In Section 5.2, we model the evaluated KB properties. Section 5.3 describes the implementations of our EL systems. Section 5.4 analyzes the biomedical data set CalbC that is used in our evaluation. Section 5.5 presents experiments in form of an in-depth evaluation. Finally, we conclude the chapter in Section 5.6.

### 5.1 Introduction

KBs represent an important aspect in EL systems by defining the basic conditions. These include the underlying domain, the specific set of entities and the entity information that can be leveraged for EL. A robust EL system, however, should be able to achieve consistent results on various domains, with a large number of entities and with a low quantity and poor quality of entity definitions (cf. Chapter 4). Basically, all these Consistency criteria refer to content-related KB properties. So far, it is unclear how and to which extent content-related KB properties influence EL results in general.

In this chapter, we pose the following research question:

**Research Question:** *How and to which extent do content-related KB properties influence EL results?*

To answer this question, we select the following three crucial KB properties whose influences are investigated throughout this chapter:

- Entity format, i.e., the way entities are described, that is intensionally (i.e., logical representations like descriptions) or extensionally (i.e., through instances and usage).
- User data, i.e., quantity and quality of externally disambiguated entities within entity-annotated documents.
- Quantity and heterogeneity of entities to disambiguate, i.e., the number and size of different domains in a KB.

To evaluate these KB properties, we focus on the biomedical domain, which is extensively represented by several large data sets and KBs. Moreover, the problems of missing user data and large-scale and heterogeneous KBs are particularly relevant and present in this specific domain. Generally, biomedical EL is a challenging task due to a considerable extent of ambiguity and, thus, has attained much attention in research in the last decade [Zwi15b].

In terms of EL approaches, we implemented three Learning-To-Rank-based (LTR) algorithms. Two approaches rely on intensional and extensional entity definitions, respectively. With our third and federated approach, we investigate whether we can further improve EL results by leveraging the knowledge from different entity formats, such as intensional and extensional entity definitions. To this end, our federated approach combines the result lists of both single approaches by means of LTR.

Overall, our **contributions** in this chapter can be summarized as follows:

- We provide a systematic evaluation of (biomedical) EL with respect to the entity format, user data and the quantity and heterogeneity of entities.
- We show that the choice of the entity format, which is used to attain the best EL results, strongly depends on the amount of available user data.
- We show that the entity format strongly affects EL results with large-scale and heterogeneous KBs.
- We show that all evaluated approaches are robust against a moderate amount of noise in user data.
- We show that by using a federated approach, which leverages both entity formats, the Consistency of EL systems can be improved significantly.

## 5.2 Modeling Knowledge Base Properties

In the following, we specify and model the KB properties entity format (Section 5.2.1), user data (Section 5.2.2), and quantity and heterogeneity of entities (Section 5.2.3) in the context of this chapter.

### 5.2.1 Modeling Entity Format

Generally, an entity can be defined intensionally, i.e., through a description, or extensionally, i.e., through instances and usage [Ogd23]. Intensional definitions can be understood as a thesaurus or logical representation of an entity, as it is provided by Linked Open Data

(LOD) repositories. In contrast, extensional definitions resemble information on the usage context of an entity, as it is provided by manually or automatically entity-annotated documents. We model these entity formats as an entity-centric or document-centric KB, equally to as defined in Definition 2.4 and 2.5 in Chapter 2 (cf. Figure 5.1).

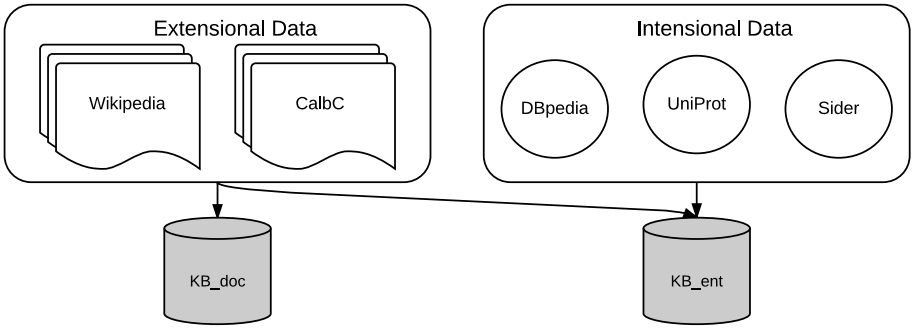



**Figure 5.1:** Modeling entity format in form of an entity-centric and document-centric KB

The edge between extensional data and entity-centric KB illustrates that an entity-centric KB may store extensional entity data. For instance, surface forms or synonyms are often extracted from document-centric KBs to improve entity-centric EL.

However, to facilitate the usage of different entity-centric and document-centric KBs, we explicitly create our own EL indexes in the respective formats. Entries in our entity-centric KBs contain standard attributes that are typically exhibited by all entities regardless of the underlying domain, such as ID, name, synonyms, description, link to web resource:

$$e_j = (ID, Name, Synonyms, Description, Link, Occurrences, Co\text{-}Occurrences) \quad (5.1)$$

Moreover, we store surface forms for each entity and, if available, the number how often the respective entity has been annotated with a specific surface form (field *Occurrences*). Additionally, for each entity $e_j$, we store a set of surface forms of other entities that have been annotated in a specific context range together with entity $e_j$ (field *Co-Occurrences*). In our experiments, we extract these information from the underlying document-centric KB. Table 5.1 shows an example of a TRNA-protein stored in our index that represents an entity-centric KB entry.

In contrast, the general structure of an entry in our document-centric index is denoted as follows:

$$d_k = (ID, Titleandtext, Annotations) \quad (5.2)$$

In addition to the entire document (field *Titleandtext*), we store all available entity annotations (i.e., surface forms and target entities) of a document in the field *Annotations*. The field *ID* depicts a unique document identifier. Table 5.2 shows an extract of a biomedical document stored in our document-centric index.

**Table 5.1:** Example of an entity-centric KB entry in our index

| Field | Content |
|---|---|
| ID | UNQ9A741 |
| Name | Phenylalanyl-tRNA–protein transferase |
| Synonyms | Leucyltransferase |
| Description | Functions in the N-end rule pathway of protein degradation where it conjugates Leu, Phe and, less efficiently, Met from aminoacyl-tRNAs to the N-termini of proteins containing... |
| Link | http://www.uniprot.org/uniprot/Q9A741 |
| Occurrences | aat:::3 |
| Co-Occurrences | substrate:::3, Leu:::6, Phe:::6 |

**Table 5.2:** Example of a document-centric KB entry in our index

| Field | Content |
|---|---|
| ID | 174996 |
| Titleandtext | Antibody therapy for treatment of multiple myeloma. Monoclonal antibody therapy antibody therapy has emerged as a viable treatment option for patients with... |
| Annotations | Myeloma::43::50::diso:umls:C0026764:T191:diso |

### 5.2.2 Modeling User Data

In our chapter, the set of all (user) annotations in natural language documents is called user data. A user annotation consists of a textual representation $m_i$, the surface form, and an **entity set** $\boldsymbol{t^i}$ with $t^i_j \in \boldsymbol{t^i}$ that is referred by surface form $m_i$. With this definition, we allow a surface form referring to multiple entities since a combination of multiple KBs can result to multiple entities being correct. In our case of using LOD resources, the correct entities are typically connected via *sameAs* relation. Example 5.1 shows an example user data annotation of surface form 'H1N1', with *id* denoting an entity's resource identifier:

**Example 5.1.**
...WHO declared <e id="UMLS:C1615607:T005:diso">H1N1</e>influenza...

User data is basically contained in document-centric KBs. However, valuable information from user data is often extracted and stored in entity-centric KBs. In our work, we assume that user data in form of a-priori entity-annotated documents is readily available and provided by the underlying document-centric KBs (cf. Section 5.4).

### 5.2.3 Modeling Large-Scale and Heterogeneous Knowledge Bases

The quantity is closely related to the heterogeneity of a KB. Increasing the heterogeneity within a KB is caused by adding entities from other domains. Hence, we distinguish between an *intra-specific* domain extension and an *inter-specific* domain extension. An

intra-specific domain extension describes a KB enrichment with entities or documents from the same domain. In our case, we add entities and documents from the biomedical domain (e.g., a gene database). In contrast, a KB enrichment with documents or entities from other domains (e.g., DBpedia/Wikipedia) describes an inter-specific domain extension.

## 5.3 Approaches

In the following, we present three ranking-based EL systems to investigate our defined KB properties, namely the entity format, user data, and the quantity and heterogeneity of entities to disambiguate. More specifically, we describe EL approaches for entity-centric and document-centric KBs. Our third and federated approach leverages different entity formats to explicitly incorporate the knowledge from intensional and extensional entity definitions (i.e., entity-centric and document-centric KBs). In order to let our approaches weight our underlying features according to their relevance and importance in each experiment, we decided to use LTR-based EL methods. Figure 5.2 provides an overview of our systems.

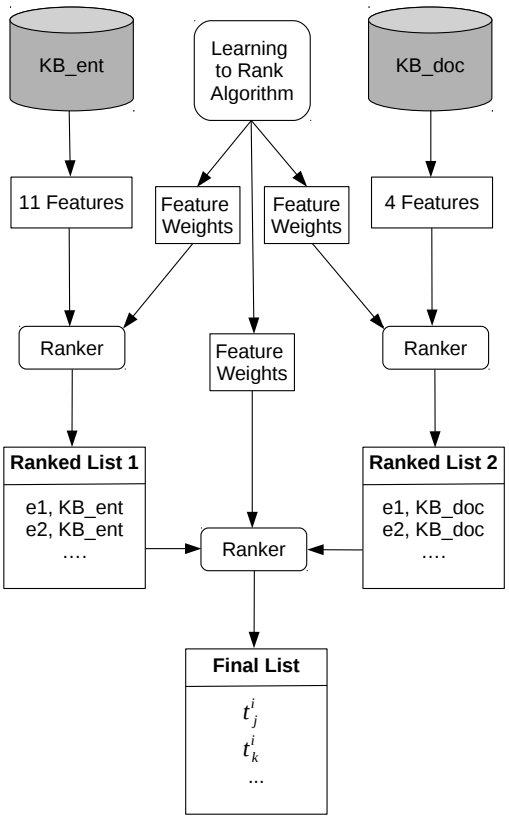



**Figure 5.2:** Overview of our entity-centric, document-centric and federated EL approaches. Our entity-centric and document-centric approaches independently link entities by means of Learning to Rank. The outcomes of both approaches are then weighted and aggregated to create a federated result list.

In contrast to our EL definition in Section 2.1.1, our EL algorithms return a ranked list of relevant entities for a surface form $m_i$ instead of a single entity assignment $t^i_j$. We allow

multiple assignments due to the following two reasons: First, our KBs contain different entity identifiers while representing the same entity. This typically occurs if multiple external knowledge sources are combined to generate a large-scale KB. Second, our data set CalbC contains surface forms that refer to multiple entities (cf. Section 5.4). Anyway, given a surface form $m_i$, its surrounding textual context $c_i^\lambda$ ($\lambda$ denotes the number of words in front of and after surface form $m_i$) and the set of possible target entities $\Omega$, each algorithm returns a ranked list $R_i$ of entities in descending score order for surface form $m_i$:

$$R_i = rank(m_i, c_i^\lambda, \Omega) \tag{5.3}$$

In the following sections, we present our entity-centric EL approach (Section 5.3.1), document-centric EL approach (Section 5.3.2) and federated EL approach (Section 5.3.3). In Section 5.3.4, we present the underlying feature set utilized by our approaches.

### 5.3.1 Entity Linking Approach for Entity-Centric Knowledge Bases

The first step in our entity-centric EL approach is to generate a set of candidate entities $\Omega_i$ for a surface form $m_i$. For this purpose, we use the Jaro-Winkler distance [Win90], which is designed and best suited for short strings such as (gene) names. All entities whose labels provide a Jaro-Winkler distance of $> 0.5$ concerning the surface form remain as candidate entities in the next step. Next, we rank all candidate entities by using a linear combination of a weighted feature set $\phi(m_i, c_i^\lambda, e_j)$ to compute a score $S_j^{ent}$ for each entity $e_j$:

$$S_j^{ent} = w^\intercal \phi(m_i, c_i^\lambda, e_j) \tag{5.4}$$

Variable $w$ denotes the weight vector for our feature set and $\phi(m_i, c_i^\lambda, e_j)$ denotes the respective feature vector. The EL result $R_i$ consists of the top-$n$ scored entities.

Depending on the used feature set, our entity-centric approach may also represent a **federated approach** to a certain degree. For instance, a typical feature for entity-centric approaches is the Sense Prior, which relies on extensional entity data (e.g., user data in document-centric KBs). In our experiments, we analyze the results of our entity-centric approach with and without leveraging user data.

### 5.3.2 Entity Linking Approach for Document-Centric Knowledge Bases

Our document-centric EL algorithm is similar to a $k$-Nearest-Neighbor classification using majority voting. First, we obtain a predefined number $\tau$ of relevant documents using the ranking function as defined in Equation 5.4 with a different feature set (cf. Section 5.3.4). A relevant document should contain similar content as given by surface form $m_i$ and its surrounding context $c_i^\lambda$. The second step encompasses the classification step. We compute the score $S_j^{doc}$ for all referenced entities in our queried document set $Q_\tau$:

$$S_j^{doc} = \sum_{l}^{|Q_\tau|} p(e_j|d_l) \tag{5.5}$$

Probability $p(e_j|d_l)$ denotes the probability of entity $e_j$ occurring in document $d_l$ (with reference to all documents in $KB_{doc}$). We estimate the probabilities as follows:

$$p(e_j|d_l) = \frac{count_{d_l}(e_j)}{\sum_{e_k \in \Omega} count_{d_l}(e_k)} \tag{5.6}$$

The function $count_{d_l}(e_j)$ returns how often entity $e_j$ is annotated in document $d_l$. Again, the result list $R_i$ consists of the top-$n$ scored entities. The quality of the results strongly depends on the number of annotated entities in the document set. Generally, when using a document-centric KB, user data must be available. Detailed experiments can be found in our evaluation Section 5.5.

### 5.3.3 Federated Approach

Our federated EL approach fully leverages both types of entity formats (i.e., entity-centric and document-centric KBs). Basically, we re-rank the entities located in the result lists $R^{ent}_{i,l}$ and $R^{doc}_{i,l}$ of our entity-centric and document-centric algorithms by means of LTR, which serves as supervised ensemble ranker. The additional variables *ent* and *doc* denote the type of the KB and parameter $l$ denotes the length of the respective approach's result list.

Overall, we compute a new score $S^{com}_j$ for every entity located in $R^{ent}_{i,l}$ and $R^{doc}_{i,l}$ and create a new result list. Therefore, we first define an entity set $Y$ that contains all entities of $R^{ent}_{i,l}$ and $R^{doc}_{i,l}$: $Y = R^{ent}_{i,l} \cup R^{doc}_{i,l}$. Further, we compute the final score $S^{com}_j$:

$$S^{com}_j = w^\intercal \phi(e_j), \text{ with } e_j \in Y \tag{5.7}$$

Similar to Equation 5.4, variable $w$ denotes the weight vector of our feature set and function $\phi(e_j)$ represents the feature vector of entity $e_j$. Instead of utilizing standard features computed based on a surface form - candidate entity pair, we exclusively use features related to the results achieved by the single approaches. These are two features representing the entity scores $S^{ent}_j, S^{doc}_j$ attained with our entity-centric and document-centric EL approaches (cf. Equation 5.4 and 5.5). We also leverage two features that describe the probability of the entity-centric or document-centric approach retrieving a correct result given the biomedical subdomain of candidate entity $e_j$. An entity may belong to one of five subdomains as given by our corpus (cf. Section 5.4). We compute the probabilities by analyzing the results of our single approaches.

Overall, we use the top-50 entities of the entity-centric and document-centric algorithms as input entities to provide a good entity repertory for the federated approach.

### 5.3.4 Feature Set

In the following, we describe our LTR feature set used for our entity-centric and document-centric EL approaches. We distinguish between three feature sets: string similarity features, prior features and evidence features (cf. Table 5.3). Our document-centric algorithm uses string similarity features only (according to the data in the KB) while the entity-centric approach employs all features to rank candidate entities.

**Table 5.3:** Overview of our Learning to Rank feature set

| Nr. | Feature |
|---|---|
| 1 | Jaro-Winkler distance between surface form and entity name |
| 2 | Cosine Sim. between TF-IDF weighted surface form and entity name vector |
| 3 | Cosine Sim. between TF-IDF weighted surface form and entity description vector |
| 4 | Cosine Sim. between TF-IDF weighted context and entity name vector |
| 5 | Cosine Sim. between TF-IDF weighted context and entity description vector |
| 6 | BM-25 score between surface form and entity description |
| 7 | BM-25 score between context and entity description |
| 8 | Prior: Occurrences of an entity |
| 9 | Sense Prior: Entity occurrences with a specific surface form |
| 10 | Co-occurrences: Entity-entity alignment |
| 11 | Term evidences: Entity-term alignment |

***String Similarity Features***

String similarity features are used in both EL approaches. In the entity-centric approach, we restrict our result list to those entities whose names or synonyms do not match with the surface form (i.e., candidate generation). For this purpose, we utilize the Jaro-Winkler distance [Win90], which is designed and best suited for short strings. Other features compute the similarity between the surface form and the entity name(s)/synonym(s) as well as the entity description. Additionally, we determine the similarity between the context words and the entity name(s)/synonym(s) as well as the entity description. We apply the cosine similarity of the respective TF-IDF weighted vectors (Vector Space Model) and the Okapi BM25 model (cf. Table 5.3 features 2-7) for similarity computation.

In the document-centric approach, we also use the Vector Space Model (TF-IDF) and Okapi BM25 model to search for documents with similar content as given by the surface form and context words. More specifically, we compute the cosine similarity and the BM-25 score between a given surface form and the whole textual content within a document (2 features). Moreover, we also compute the same similarities between the surrounding context of a surface form and the whole textual content within a document (2 features). An in-depth explanation of the underlying models is provided in [Man08].

***Prior Features***

Generally, some entities (i.e., *Influenza*) occur more frequently than others (i.e., *IIV3-011L gene*) in documents. Thus, these popular entities provide a higher probability to re-occur in other documents. In our work, the prior $p(e_j)$ of an entity describes the a-priori probability that an entity occurs [Res95]. A logarithm is used for this feature to damp high values. The Sense Prior $p(e_j|m_i)$ estimates the probability of seeing an entity with a given surface form. Both, Entity Prior and Sense Prior, are computed as defined in Equation 3.1 and 3.2 in Section 3.2.2.

***Evidence Features***

The *Co-occurrence* feature $Co_{e_j}$ considers context words of a surface form $m_i$ as potential surface forms for entities. Basically, we assume that surface form $m_i$'s real referent entity provides a higher probability to co-occur with potential but not yet disambiguated entities located in the surrounding context. First, we assume the context words $c_i^\lambda$ of our surface form $m_i$ to be surface forms of other entities. Parameter $\lambda$ defines the number of context words before and after the respective surface form. We compare the context words $c_i^\lambda$ to all existing surface forms provided by available user data. If a context word $w_k \in c_i^\lambda$ matches with one of these surface forms, we use this surface form's referent entity $e_l$ and compute the probability of our entity candidate $e_j$ co-occurring with $e_l$. For instance, the context word 'influenza' of surface form $m_i$ has already been used as surface form to address the entity *H1N1* in a document. Thus, entity *H1N1* describes a potential entity for our context word and we compute the probability of our entity $e_j$ co-occurring with *H1N1*. We investigate all context words $c_i^\lambda$ to compute the feature score:

$$Co_{e_j} = \sum_{w_k \in c_i^\lambda} \log(1 + \underset{e_l \in f(w_k)}{argmax}\ p(e_l|e_j)p(e_l|w_k)) \tag{5.8}$$

Function $f(w_k)$ delivers a set of entities that have been annotated in combination with the possible 'surface form' $w_k$ in other documents. We take the Sense Prior $p(e_l|w_k)$ into account to estimate the probability of surface form $w_k$ describing entity $e_l$. Further, $p(e_l|e_j)$ describes the probability of entity $e_l$ co-occurring with our candidate entity $e_j$ and is computed as follows:

$$p(e_l|e_j) = \frac{count_{e_j}^\lambda(e_l)}{\sum_{e_k \in \Omega} count_{e_j}^\lambda(e_k)} \tag{5.9}$$

The function $count_{e_j}^\lambda(e_l)$ returns the number of occurrences of entity $e_l$ in the context of $e_j$. More specifically, we analyze the user data within all underlying, annotated documents and count how often the entities $e_l$ and $e_j$ are annotated within the context range $\lambda$. Overall, we apply the logarithm to marginally improve the results.

Similar to the feature above, the *Term Evidence* feature considers probabilities of context words co-occurring with a candidate entity. For instance, the context word 'disease' is an indicator of the entity *Influenza* being correct. Our approach is similar to the entity-context model explained in Section 3.2.4 on Page 34. We compute the probabilities $p(w_k|e_j)$ of a context word $w_k \in c_i^\lambda$ of surface form $m_i$ occurring in the context of entity $e_j$:

$$p(c_i^\lambda|e_j) = \prod_{w_k \in c_i^\lambda} \frac{count_{e_j}^\lambda(w_k)}{\sum_{w_l \in T} count_{e_j}^\lambda(w_l)} \tag{5.10}$$

The function $count_{e_j}^\lambda(w_k)$ counts how often $w_k$ has been annotated in the context of entity $e_j$, with $\lambda$ denoting the context range and $T$ representing the dictionary. To this end, we analyze the user data within all underlying annotated documents.

## 5.4 Data Set

To evaluate our KB properties, we have chosen the CalbC (Collaborative Annotation of a Large Biomedical Corpus), a biomedical domain specific, document-centric KB representing a very large and silver standard text corpus annotated with biomedical entity references [Kaf12]. Overall, we use the CalbC due to the following two reasons:

- In contrast to gold standard corpora like the BioCreative (II) corpora[1], CalbC provides a huge set of annotations which perfectly suits for our evaluation purpose in terms of quantity (24 447 annotations in Biocreative II versus ≈ 120 million annotations in CalbC). It is noted that despite some annotations might being erroneous, the corpus most likely serves as predictive surrogate for a gold standard corpora [Kaf12].
- The CalbC already represents a document-centric KB comprising biomedical documents annotated with biomedical entities. A bulk of the annotated entities can be linked to their respective entries in the LOD cloud whose data sets can be interpreted as entity-centric KBs.

Table 5.4 shows some basic statistics about both CalbC subcorpora, CalbCSmall and CalbCBig, whereby both corpora are disjunct in terms of their appearing documents. Although the number of entity annotations is more than two times higher than in CalbCSmall, CalbCBig provides less distinct entity references. Additionally, it is important to mention that in contrast to other corpora like Wikipedia, an annotation in CalbC may comprise more than one entity annotation. A rich taxonomy and classification system is responsible for 9 entity annotations on average per surface form. In our work, we asses this behavior with the possibility of having more valid solutions per surface form.

**Table 5.4:** Statistics of the CalbCSmall and CalbCBig corpora

| | **CalbCSmall** | **CalbCBig** |
|---|---:|---:|
| Documents | 174 999 | 714 282 |
| Document Type | MEDLINE abstract | MEDLINE abstract |
| Surface Forms | 2 548 900 | 10 304 172 |
| Distinct Surface Forms | 50 725 | 101 439 |
| Entities | 37 309 221 | 96 526 575 |
| Distinct Entities | 453 352 | 308 644 |
| Used Distinct Entities | 265 532 | 228 744 |
| Namespaces | 14 | 16 |

Given the annotated entity set across all CalbC documents, we are able to generate an entity-centric KB by gathering information from the respective LOD repositories. For each user annotation we are able to create a link to the respective RDF resource. Because some namespaces are not publicly available, we did not consider those entities during the parsing process. Instead, we focus on the four major namespaces UMLS, Disease (is contained in

1 http://www.biocreative.org/news/biocreative-ii/, last accessed on 2016-11-28

UMLS), Uniprot and EntrezGene, which constitute the majority of annotated entities in both CalbC data sets. The UMLS dataset is a combination of many health and biomedical vocabularies, whereas Uniprot provides high-quality resources of protein sequences and function information. EntrezGene exclusively comprises gene-specific information.

## 5.5 Evaluation

In our evaluation, we provide a systematic evaluation of biomedical EL with respect to the entity format, user data and the quantity and heterogeneity of entities. First, we describe the experimental setup in Section 5.5.1. Second, we investigate the user data influence on our document-centric approach. Here, we analyze how different scales of user data and different values of parameter $\tau$ (i.e., the number of documents for classification) affect the EL results (Section 5.5.2). Third, we compare our entity-centric, document-centric and federated approaches in the context of different amounts of user data (Section 5.5.3). In this context, we emphasize that our intention is not to compare our approach to other approaches because most publicly available biomedical entity annotators do not return a ranked list (e.g., NCBO annotator[1]), which is a key factor in our evaluation. Fourth, we evaluate how the entity format and user data influence the accuracy with large-scale and heterogeneous KBs (Section 5.5.4). Fifth, we analyze how EL results evolve after adding different degrees of erroneous user data (Section 5.5.5).

### 5.5.1 Experimental Setup

Our approaches are implemented in Java with all queries being executed with Apache Lucene 6.0.1[2]. For the LTR algorithm, we chose Sofia-ml[3], a machine learning framework providing algorithms for massive data sets [Joa02]. We describe our single results with a set of comprehensive measures, including mean reciprocal rank (MRR), recall and mean average precision (MAP), which are averaged over 5-fold cross validation runs. The reciprocal rank is the multiplicative inverse of the rank of the first correct result in a result set. The average precision denotes the average of all precision@$n$ values of a single EL task. A precision@$n$ value is computed at every correct hit $n$ in the result set [Man08]. Similar to search engines, correct evaluation results should appear at the top of the result list. For this very reason, a high reciprocal rank in combination with a strong recall are desirable. On the other hand, we relinquish the usage of the precision measure because a fixed number of results is returned by our EL system. Instead, the MAP computes the precision at each correct hit in the result list.

In terms of parameters, we will only present the most important ones. The context length $\lambda$ affects the number of words in both directions, before and after the corresponding surface form. We determined a context length of 50 words. More words worsen the results in all experiments. By using Lucene's TF-IDF score, it must be noted that Lucene's default TF-IDF score also takes internal parameters like term boosting and coordination factor into account. Our entity-centric approach always uses fuzzy queries to query the

1 http://bioportal.bioontology.org/annotator, last accessed on 2016-11-28
2 http://lucene.apache.org/, last accessed on 2016-11-28
3 http://code.google.com/p/sofia-ml/, last accessed on 2016-11-28

surface forms and term queries to query the surrounding context. Fuzzy queries match terms with a maximal edit distance of 2. The document-centric approach always uses term queries for surface forms and context queries. To determine the best parameter $\tau$ for our document-centric algorithm, i.e., the number of documents used for classification, we perform an in-depth parameter study in Section 5.5.2. Finally, our overall result lists are trimmed to 10 entities per query to provide a good relation between recall and precision.

We emphasize, that we consistently use the CalbCSmall corpus in our evaluation. The CalbCBig data set serves for scalability experiments as conducted in Section 5.5.4.

### 5.5.2 Influence of User Data on Document-Centric Entity Linking

In this section, we investigate the influence of user data on our document-centric approach. More specifically, we analyze how the number of documents used to classify entities (i.e., parameter $\tau$) and the number of annotations within these documents influence the results on the CalbC data set. In the default experiment configuration, our approach uses all annotations in CalbC (i.e., 100% user data). For all other scales, the KB and probabilities (needed to compute Equation 5.5) were reconstructed and computed accordingly. To create our KB with a specific fraction of the original user data (for instance 0.1% or 25%), we stored a user annotation of a CalbC document with the respective probability in our KB. For instance, 25% user data refers to an annotation is kept in the corpus with a probability of $p = 0.25$. Figure 5.3 and Table 5.5 show an overview of our results. The plot's x-axis starts at 0.1% due to the necessity of user data in our approach. The abbreviation U.D. x denotes that x% of the overall available amount of user data is used in the experiment.

Basically, all result values improve with an increase of user data regardless of $\tau$. Poor

**Table 5.5:** Results of our document-centric EL approach with various amounts of user data

| Measure | Parameter $\tau$ | U.D. 0.1 | U.D. 1.0 | U.D. 20.0 | U.D. 100.0 |
|---|---|---|---|---|---|
| MRR | 100 | 0.355 | 0.641 | **0.792** | **0.806** |
| | 250 | 0.425 | 0.708 | 0.788 | 0.790 |
| | 750 | 0.534 | 0.721 | 0.782 | 0.769 |
| | 1500 | 0.571 | 0.722 | 0.767 | 0.755 |
| | 2500 | **0.624** | **0.739** | 0.735 | 0.664 |
| Recall | 100 | 0.231 | 0.511 | **0.753** | **0.767** |
| | 250 | 0.289 | 0.553 | 0.745 | 0.746 |
| | 750 | 0.388 | 0.584 | 0.728 | 0.733 |
| | 1500 | 0.422 | 0.589 | 0.712 | 0.717 |
| | 2500 | **0.464** | **0.592** | 0.694 | 0.663 |
| MAP | 100 | 0.163 | 0.410 | **0.621** | **0.635** |
| | 250 | 0.216 | 0.465 | 0.611 | 0.618 |
| | 750 | 0.299 | 0.487 | 0.596 | 0.601 |
| | 1500 | 0.337 | 0.493 | 0.593 | 0.595 |
| | 2500 | **0.359** | **0.507** | 0.588 | 0.569 |

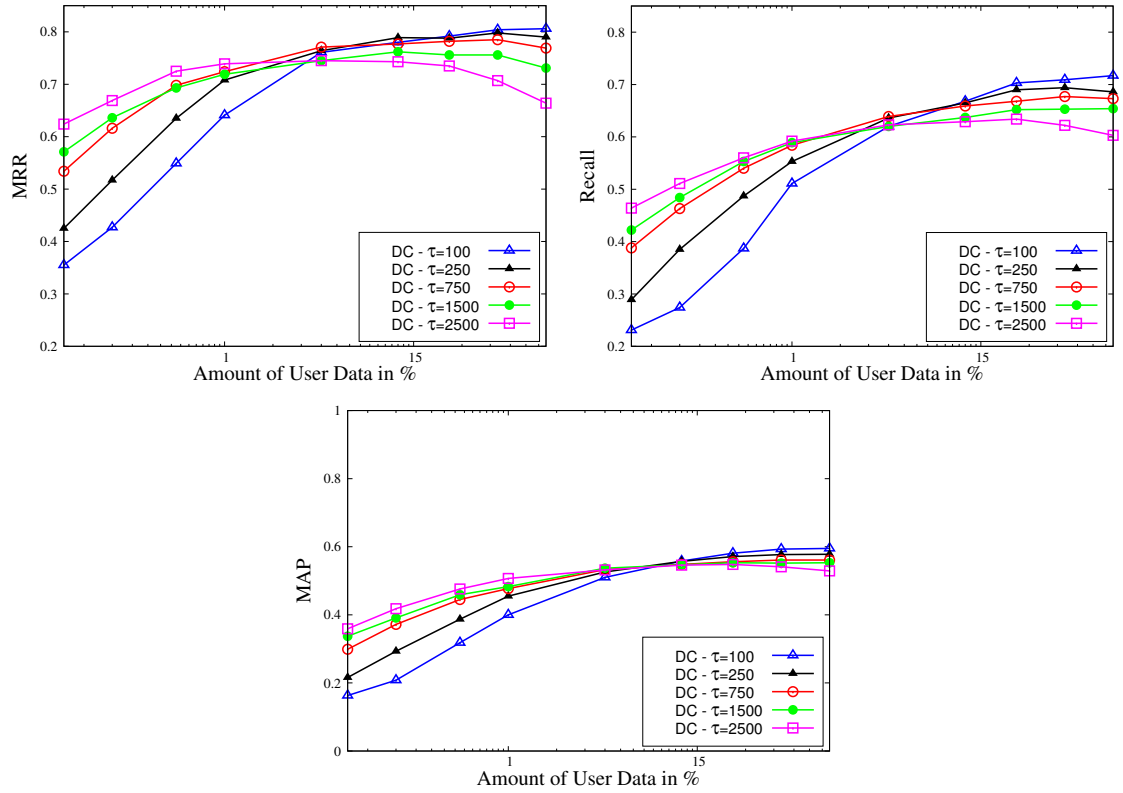



**Figure 5.3:** Various scales of user data with our document-centric EL algorithm. Parameter $\tau$ denotes the number of documents used for classification.

MRR, recall and MAP values (between 0.15 and 0.35) when using $\tau = 100$ and 0.1% user data indicate that the amount of user data is absolutely insufficient to provide enough evidence for good EL results. Consequently, using more documents (e.g., $\tau = 1500$) with few annotations per document improves the results. The story looks different if too many annotations are available. This is the case when we choose high values for parameter $\tau$ and the selected documents contain a high number of annotations on average. While the recall and MAP values nearly stay constant with $\tau = 1500$ and 100% user data, the MRR significantly drops about 8 percentage points due to too much noise compared to $\tau = 1500$ and 12% user data. The results achieved with $\tau = 2500$ confirm that richly annotated documents (100% user data) in combination with many documents (high values for $\tau$) mitigate the EL results.

In the following, we dig deeper into the previous outcomes and investigate the EL results when we use a fixed number of annotations in the classification step of our algorithm. Therefore, we introduce a latent parameter $\Lambda$, which specifies a fixed number of annotations across all documents used in the classification step. Now, our parameter $\tau$ depends on the number of annotations in the documents used for classification and parameter $\Lambda$. A first experimental run showed an improvement, but did not always provide the best results. This is the case if we set $\Lambda$ to a low value and the documents used for classification are

long while providing many annotations (resulting in very few documents for classification overall). We also noted decreasing results when we used higher values for $\Lambda$ and short documents (resulting in a huge set of annotations distributed across many documents). We omit an in-depth elaboration of this parameter due to its marginal improvements.

In **summary**, we state that the number of documents used for classification (parameter $\tau$) and the number of annotations within these documents must be well-matched to attain the best result. In the following sections, we use $\tau = 1500$ for our main experiments since it provides the best averaged results with various amounts of user data.

### 5.5.3 Comparing Entity Linking Approaches with Different Amounts of User Data

In this section, we investigate the influence and effects of the entity format onto EL accuracy. Furthermore, we compare our approaches with different scales of user data. To this end, we re-created our models with various fractions of user data (cf. Section 5.5.2).

Table 5.6 shows an overview of the results achieved by our algorithms with various amounts of user data (i.e., 0.1%, 1%, 20% and 100% user data). For a better estimation we can say that 1% of user data approximately corresponds to 1 annotation per entity on average. We compare our entity-centric ($EC$), document-centric ($DC$) and federated approaches while user data must be available for the document-centric and federated approaches. Figure 5.4 shows the MRR, recall and MAP values of our approaches. We note that the plot's x-axis starts at 0.1% again due to its logarithmic scale to improve visualization and its necessity of user data for document-centric and federated EL. Again, we use U.D. x to denote the amount of user data in this configuration.

**Table 5.6:** MRR, recall and MAP values of our entity-centric, document-centric and federated EL approaches with various amounts of user data

| Measure | Approach | U.D. 0 | U.D. 0.1 | U.D. 1.0 | U.D. 20.0 | U.D. 100.0 |
|---|---|---|---|---|---|---|
| MRR | EC | **0.367** | 0.447 | 0.702 | 0.855 | 0.880 |
| | DC | - | 0.571 | 0.719 | 0.756 | 0.755 |
| | Federated | - | **0.585** | **0.739** | **0.923** | **0.927** |
| Recall | EC | **0.253** | 0.299 | 0.562 | **0.742** | **0.767** |
| | DC | - | **0.422** | **0.589** | 0.718 | 0.717 |
| | Federated | - | 0.373 | 0.580 | 0.716 | 0.718 |
| MAP | EC | **0.257** | 0.284 | **0.509** | 0.681 | 0.707 |
| | DC | - | **0.337** | 0.478 | 0.595 | 0.595 |
| | Federated | - | 0.279 | 0.508 | **0.685** | **0.709** |

In the following discussion, we assume that a significant amount of user data is available (i.e., all annotations in CalbC). In this setting, our entity-centric approach achieves a high MRR (0.880), recall (0.767) and MAP (0.707) and significantly outperforms the document-centric approach in all measures. Our federated approach further improves the results and leads the entity-centric approach by 4 percentage points MRR when we

leverage all user annotations. A MRR of 0.927 shows a high level of reliability in terms of ranking a correct entity on top. In contrast, the high recall values (0.767) provided by the entity-centric approach are not transfered. Instead, the federated approach attains similar results as provided by the document-centric approach (0.718). We assume that optimizing our LTR weights with respect to recall and using additional features may overcome this deficit. Nevertheless, the MAP values of the federated approach are nearly similar to those achieved by the entity-centric approach (0.709).

Analyzing Figure 5.4 shows that the amount of user data strongly influences the MRR, recall and MAP values of our approaches. The entity-centric approach significantly outperforms the document-centric approach if enough user data is available. In contrast, we note reverse results if the amount of user data (significantly) decreases. The less user data is available, the higher is the advance of the document-centric approach. This shows that the results of our entity-centric approach strongly depend on user data. Exclusively leveraging the entity definitions in the LOD data sets in combination with our defined feature set does not lead to satisfying EL results.

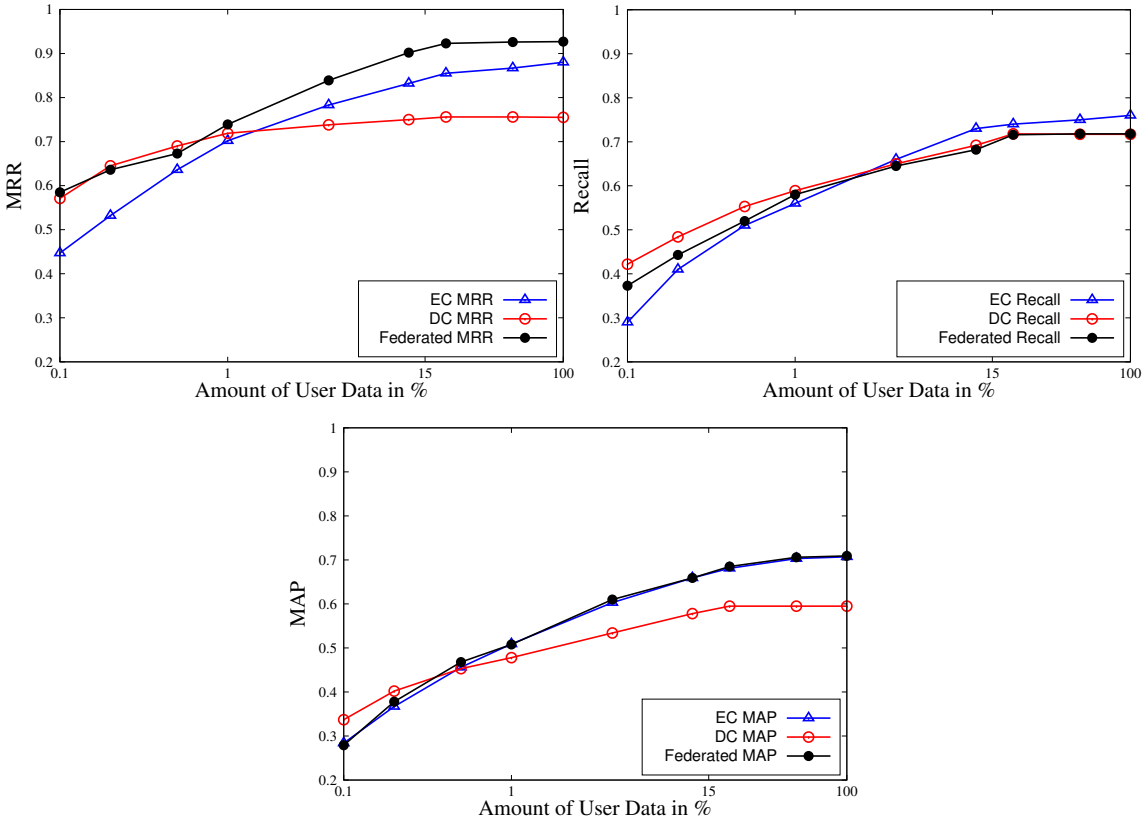



**Figure 5.4:** Results of our entity-centric, document-centric and federated EL approaches with various amounts of user data

In **summary**, we state that neither our entity-centric nor our document-centric approach attains the best results with all configurations. The choice of the best entity format in terms of EL results strongly depends on the amount of available user data. For instance,

our document-centric approach performs significantly better when the amount of user data is strongly limited. In contrast, our entity-centric approach can be significantly improved by additionally leveraging extensional entity definitions in form of user data. This indicates that leveraging intensional and extensional entity definitions may significantly improve EL results, which is also affirmed by our federated approach. Overall, we recommend to leverage both entity formats in form of a federated EL approach since it provides sophisticated results with an excessive and limited amount of user data.

### 5.5.4 Knowledge Base Size and Heterogeneity

In the following, we analyze how the entity format influences EL results when the size and/or heterogeneity of the KBs is increased. To this end, we extended our entity-centric KBs $KB_{\text{ent}}$ and $KB_{\text{ent/ua/sb}}$ with additional entities. $KB_{\text{ent}}$ denotes an entity-centric KB without user data information and $KB_{\text{ent/ua/sb}}$ denotes the enrichment of the entity-centric KB with user data information (ua) of CalbCSmall (s), CalbCBig (b) or both (sb). The set of additional entities comprises all entities belonging to UMLS, Uniprot and/or DBpedia. For all DBpedia entities, we used the *rdfs:label* attribute as entity name and the *dbo:abstract* attribute as entity description. We enriched our document-centric KB with the CalbCBig data set (intra-specific extension) and/or Wikipedia pages (inter-specific extension). Table 5.7 shows an overview of the results before and after extending the KBs.

The entity-centric approach without user data achieves significantly worse results after increasing the number of entities (first table section). Additionally, increasing the domain heterogeneity by adding DBpedia entities further worsens the results with a result decrease of 33% (with DBpedia entities only), respectively 40% (with DBpedia, UMLS and Uniprot entities) on average.

Our entity-centric KB approach is more robust against an increase of entities and heterogeneity when we additionally leverage user-annotated documents from CalbCSmall (second table section). More specifically, the result decrease is limited to 29% instead of 40% on average. Additionally using user data from CalbCBig further improves the results by 3 percentage points on average. Despite improving results with user data, our entity-centric approach does not provide consistent results with large-scale and heterogeneous KBs. We assume that our LTR approach is not able to appropriately weight the underlying feature set in order to cope with additional entities from heterogeneous domains. It is an open question whether there exist features that suppress these negative effects.

The results of our document-centric approach remain constant when we add additional biomedical documents (third table section). We assume that the document increase does not influence the classification step (cf. Section 5.3.2). Instead, the retrieval step has a greater variety of documents to retrieve. Selecting other documents has only a minor effect on the documents' spectrum of annotated entities. An inter-specific domain extension with CalbC and Wikipedia documents leads to decreasing EL results of 11% on average. In contrast to our entity-centric approach, the document-centric approach is significantly more robust against an intra and inter-specific domain extension.

The results of our federated approach are more robust than those of our entity-centric approach (fourth table section). With the document-centric approach being robust against the document count, the accuracy decrease after increasing the heterogeneity and entity/-

document count remains smaller. More specifically, adding entities and documents from UMLS, Uniprot, DBpedia, CalbCBig and Wikipedia results in an average result decrease of $\approx 20\%$ compared to our default settings without entity and document extensions.

**Table 5.7:** Results after increasing our KBs with various corpora

| Settings | Integrated KBs | MRR | Recall | MAP | #Ent/#Docs |
|---|---|---|---|---|---|
| $KB_{ent,\ intra}$ | - | 0.367 | 0.253 | 0.257 | 265 532 |
| $KB_{ent,\ intra}$ | UMLS, Uniprot | 0.309 | 0.204 | 0.195 | 32 407 960 |
| $KB_{ent,\ inter}$ | DBpedia | 0.256 | 0.177 | 0.183 | 4 643 509 |
| $KB_{ent,\ inter}$ | UMLS, Uniprot, DBpedia | 0.229 | 0.140 | 0.154 | 36 785 937 |
| $KB_{ent/ua/s,\ intra}$ | - | 0.880 | 0.767 | 0.707 | 265.532 |
| $KB_{ent/ua/sb,\ intra}$ | - | 0.905 | 0.792 | 0.732 | 265 532 |
| $KB_{ent/ua/s,\ intra}$ | UMLS, Uniprot | 0.780 | 0.666 | 0.609 | 32 407 960 |
| $KB_{ent/ua/s,\ inter}$ | UMLS, Uniprot, DBpedia | 0.603 | 0.559 | 0.501 | 36 785 937 |
| $KB_{ent/ua/sb,\ inter}$ | UMLS, Uniprot, DBpedia | 0.627 | 0.580 | 0.524 | 36 785 937 |
| $KB_{doc,\ intra}$ | - | 0.755 | 0.717 | 0.595 | 174 999 |
| $KB_{doc,\ intra}$ | CalbCBig | 0.760 | 0.722 | 0.601 | 889 282 |
| $KB_{doc,\ inter}$ | CalbCBig, Wiki | 0.673 | 0.650 | 0.508 | 4 267 259 |
| $KB_{federated,\ intra}$ | - | 0.927 | 0.718 | 0.709 | 440 531 |
| $KB_{federated,\ intra}$ | CalbCBig, UMLS, Uniprot | 0.819 | 0.659 | 0.615 | 33 297 242 |
| $KB_{federated,\ inter}$ | CalbCBig, UMLS, Uniprot, DBpedia, Wiki | 0.757 | 0.601 | 0.516 | 37 675 219 |

In **summary**, we state that increasing the size and heterogeneity in KBs plays a crucial role in robust EL. As shown in our experiments, the EL results (significantly) decrease when we combine various KBs, especially when we add KBs from other domains (inter-specific domain extension). More specifically, our document-centric approach is significantly more robust against large-scale and heterogeneous KBs than the entity-centric approach (without user data) in both, inter-specific and intra-specific domain extension. Our entity-centric approach with user data and the federated approach mitigate the accuracy decrease. Overall, we recommend to leverage intensional and extensional entity definitions to mitigate the accuracy decrease after increasing the size and heterogeneity in KBs.

### 5.5.5 Noisy User Data

Available user data may contain errors caused by missing knowledge or validation errors. In the following, we investigate how additional noise in annotations influences the results of our entity-centric, document-centric and federated EL approaches. To this end, we

compare a user model created from the original annotations (as given by CalbC) to user models with different degrees of additional annotation errors. Prior research has already investigated the influence of noisy user data on LTR models [Kum11], but the effects on EL results remain unknown. However, we modified available CalbC annotations and re-created our KBs and LTR models. Therefore, we selected an annotation to be wrong with probability $p$. Instead of exchanging the annotation with a randomly selected entity annotation, we simulated user behavior by choosing a wrongly disambiguated entity. More specifically, given the entity result list of our entity-centric approach for a specific surface form, we selected a wrong entity to be replaced with the original one. Choosing a wrong entity at the top of the result list should be more likely than choosing an entity from the end. We modeled this event with a Gaussian distributed random variable $X_{m_i} \sim \mathcal{N}(1, 10)$. Our random variable $X_{m_i}$ yields positive values only since negative values are useless in terms of selecting an entity position in the result list. We exchanged the correct annotation with the wrong result that was selected by the random variable. We modified the CalbC annotations with varying degrees of noise. Figure 5.5 shows the evaluation results with 0% additional noise (as given by CalbC) to 100% noise (all annotations are wrong) attained with our entity-centric, document-centric and federated approaches.

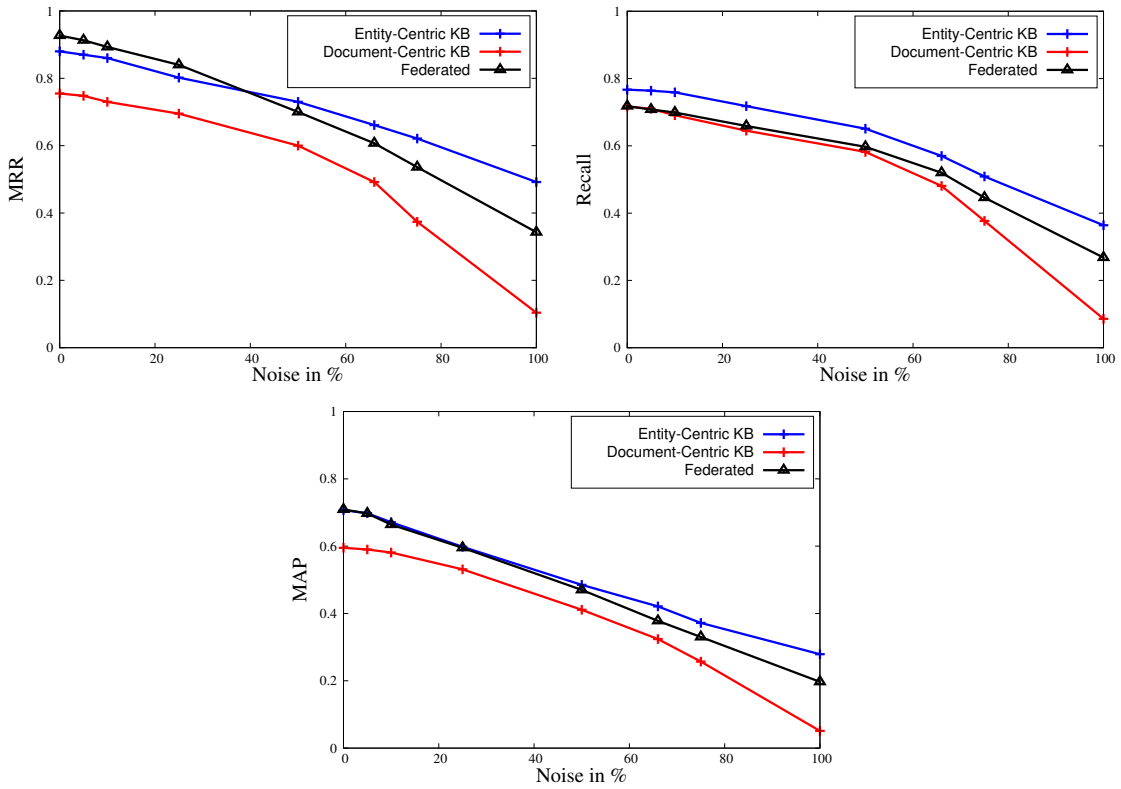



**Figure 5.5:** Influence of noise in user data on EL results

During the following result discussion, we focus on the results with a 25% noise rate. The MRR of the entity-centric and document-centric approaches shows a slight decrease of 10 percentage points with a noise rate of 25%. The federated approach tops the single

approaches as long as the noise rate remains under 33%. In all approaches, the recall decrease is about 5 percentage points with 25% noise. Basically, the recall values remain high as long as the noise rate does not exceed 66%. In terms of MAP values, the values of our entity-centric and federated approaches continuously decrease almost linearly from 0% to 100% noise. However, a decrease of up to 12 percentage points with 25% noise in all approaches shows that the MAP values are slightly more affected by noisy user data.

In **summary**, we note that all approaches are robust against little noise in user data. Assuming that the amount of erroneously annotated data is about one third or less, we emphasize that all EL approaches are robust and still provide satisfying results.

## 5.6 Conclusion and Discussion

In this chapter, we provided a systematic evaluation of (biomedical) EL with respect to three major KB properties that are particularly relevant in the context of consistent EL. These are the entity format (i.e., intensional and extensional entity definitions as provided by entity-centric or document-centric KBs), user data (i.e., entity-annotated documents) and the quantity and heterogeneity of entities. To this end, we implemented three LTR-based EL systems to model different entity formats. Two approaches rely on entity-centric and documents-centric KBs, respectively. Our third and federated approach leverages both, intensional and extensional entity definitions, by combining the linking results of both single approaches.

Our evaluation revealed that all three KB properties significantly influence EL results. More specifically, the choice of the entity format that is used to attain the best EL results strongly depends on the amount of available user data. Our federated approach achieves the best results on average. The entity format is also crucial in terms of providing Consistency with large-scale and heterogeneous KBs. Our federated approach provides significantly better results with large-scale and heterogeneous KBs than our entity-centric approach. As a result, we suggest to utilize either a document-centric or a federated approach to cope with these kind of KBs. In terms of noisy user data, all three EL approaches are robust against a moderate amount of noise in user data.

Overall, we suggest to leverage both types of entity definitions, intensional and extensional entity definitions, in an EL approach. This can be achieved by using a federated approach that combines the results of entity-centric and document-centric EL approaches as shown in this chapter. Another possibility is, for example, appropriate feature design in an entity-centric approach. For instance, our entity-centric EL approach also contains features based on extensional entity definitions (i.e., user data). As indicated in the conducted experiments, these extensional entity definitions can strongly improve entity-centric EL in terms of EL accuracy and robustness.

Regarding future, robust EL systems, we showed that EL with an entity-centric KB (as it is often the case in very specialized domains where available user data is limited) can be significantly improved even with a low amount of user data. Considering the results of our user data experiments, we suggest to automatically annotate documents with an entity annotation system since a moderate degree of noisy user data does not significantly decrease EL results. Nevertheless, an analysis of the integrated KBs, the size and the degree of KB heterogeneity as well as the amount of available user data must be considered

to spot the potential problem areas and adapt the underlying EL systems accordingly.

A **limitation** of our evaluation is the choice of three very specific EL approaches that are tailored toward the respective KBs. In Chapter 3, we showed the diversity of EL algorithms and approaches that have been proposed in the literature. Choosing another set of EL algorithms in our evaluation would have led to different EL results. However, we strongly assume that the key messages remain the same.

# CHAPTER 6

## Entity Relatedness

In this chapter, we present a new state-of-the-art entity relatedness measure for collective Entity Linking (EL), which provides Structural Robustness and Consistency. Our measure is based on entity embeddings (i.e., low-dimensional vectors), which are trained with Word2Vec, a set of popular algorithms to create word embeddings. We propose two algorithms to create appropriate Word2Vec input corpora from entity-centric and document-centric knowledge bases (KBs). In terms of evaluation, we integrated our new entity relatedness measure in a graph-based, baseline algorithm for collective EL and compare our approach to existing state-of-the-art approaches. In our experiments, we (significantly) outperform all other publicly available, state-of-the-art, collective EL approaches on entity-centric and document-centric KB. This chapter partially covers the ideas, findings and materials published in the work [Zwi16a].

The remainder of this chapter is structured as follows: After introducing the chapter in Section 6.1, we present our new entity relatedness measure in Section 6.2. Section 6.3 describes our graph-based, collective baseline approach and Section 6.4 presents the data sets used in our evaluation. In our evaluation in Section 6.5, we present the results of our relatedness measure in detail. Finally, Section 6.6 concludes this chapter.

### 6.1 Introduction

Measuring the relatedness between an entity pair is an essential step in collective EL approaches. It has been shown that this kind of technique leads to state-of-the-art results when linking surface forms within textual documents, tweets or tables. However, in the literature, a plethora of different entity relatedness measurements has been proposed (cf. Section 3.2.5), but most approaches are tailored toward a specific KB. In order to create a robust EL system, we need to employ an entity relatedness measure that can be easily computed on entity-centric and document-centric KBs. Moreover, the relatedness measure should ideally be robust against a low quantity and poor quality of entity data, such as entity annotations or relations between entities.

In this chapter, we pose the following research question:

> **Research Question:** *Which entity relatedness measure provides Structural Robustness and Consistency while achieving state-of-the-art results in EL systems?*

In this chapter, we present a new entity relatedness measure for collective EL that is based on Word2Vec [Mik13a; Mik13b]. Word2Vec is a set of unsupervised algorithms for creating word embeddings (i.e., real-valued n-dimensional vectors capturing the semantics

of words) from (textual) documents. In our work, the relatedness between two candidate entities is determined by computing the similarity of the respective entity embeddings. To compute these embeddings, we need to create a training corpora first, which serves as input for the Word2Vec algorithm. In the context of this work, we propose algorithms to generate appropriate training corpora for entity-centric and document-centric KBs.

To evaluate our approach, we might re-implement all other existing state-of-the-art techniques that can be applied to both kinds of KBs and evaluate them within the same EL approach. Unfortunately, this is extremely time-consuming since a lot of approaches were proposed in the literature and implementation details are often missing due to insufficient descriptions. As a consequence, we decided to separately evaluate our new proposed entity relatedness measure on a very simple but popular, collective, graph-based approach. We compare our approach to other more sophisticated approaches to show that our entity relatedness measure achieves state-of-the-art EL results on different KBs.

Overall, our **contributions** in this chapter can be summarized as follows:

- We present a new KB-agnostic, state-of-the art entity relatedness measure based on semantic entity embeddings.
- We show that our entity relatedness measure integrated in a collective, baseline EL approach achieves state-of-the-art EL results on most data sets.
- We show that our entity relatedness measure provides Structural Robustness and Consistency in terms of poor quality entity definitions in entity-centric and document-centric KBs.

## 6.2 Entity Relatedness Based on Semantic Embeddings

Embedding is the collective name for a set of models and feature learning techniques where any concepts are mapped to vectors of real numbers in a low-dimensional space. This has already been well researched for words in literature [Ben03; Mik13a; Pen14]. In this chapter, we focus on Word2Vec, the current state-of-the-art technique to produce word embeddings, and show how entity embeddings can be generated for different types of source KBs. First, we briefly introduce Word2Vec in Section 6.2.1. Second, we propose how to create entity embeddings suited for entity relatedness computation on the basis of entity-centric and document-centric KBs in Section 6.2.2.

### 6.2.1 Word2Vec

Word2Vec is a group of state-of-the-art neural network language models to create word embeddings from (textual) documents unsupervised, initially presented by Mikolov et al. [Mik13a; Mik13b]. To train these embeddings, Word2Vec uses a two-layer neural network to process non-labeled documents. The neural network architecture is based either on the continuous bag-of-words (CBOW) or the skip-gram architecture.

When using the CBOW model, the task aims to predict a word given its surrounding context. More specifically, the input to the model could be a set of context words $c_h = \{w_{h-k},...,w_{h-1},w_{h+1},...,w_{h+k}\}$ that denote the preceding and following words of the current word $w_h$. The output of the network is a Multinomial distribution over the dictionary with a probability for each word being the correct word. The training objective

is to maximize the conditional probability of observing the actual output word $w_h$, given its context words $c_h$:

$$max\ p(w_h|c_h) = max\ \frac{exp({v'_{w_h}}^{\top}\ v_{c_h})}{\sum_{l=1}^{T} exp({v'_{w_l}}^{\top}\ v_{c_h})}, \tag{6.1}$$

where $v_{c_h}$ denotes the averaged weight vector of the context words using the weight matrix between the hidden layer and the output layer (i.e., softmax weights). Further, $v'_{w_h}$ is the weight vector of the target word $w_h$ using the weight matrix between the input layer and the hidden layer (i.e., word weights).

The skip-gram model works vice-versa. The input to the model is a word $w_h$ and the neural network predicts its surrounding context words $c_h$. Thus, instead of outputting one Multinomial distribution overall, the network outputs $|c_h|$ Multinomial distributions. The output probability for each word $w_k \in T$ is the same in each distribution:

$$p(w_k|w_h) = \frac{exp({v'_{w_k}}^{\top}\ v_{w_h})}{\sum_{l=1}^{T} exp({v'_{w_l}}^{\top}\ v_{w_h})} \tag{6.2}$$

The training objective of the skip-gram model is to maximize the conditional probability of observing the set of context words $c_h$ given the word $w_h$: $max\ p(c_h|w_h)$.

An important property of Word2Vec is that it groups the vectors of similar concepts together in the vector space. If enough data is used for training, Word2Vec makes highly accurate guesses about a word's meaning based on its past appearances. Figure 6.1 shows a low-dimensional projection of skip-gram vectors. It illustrates the ability of the model to automatically organize concepts and implicitly learn the relationships.

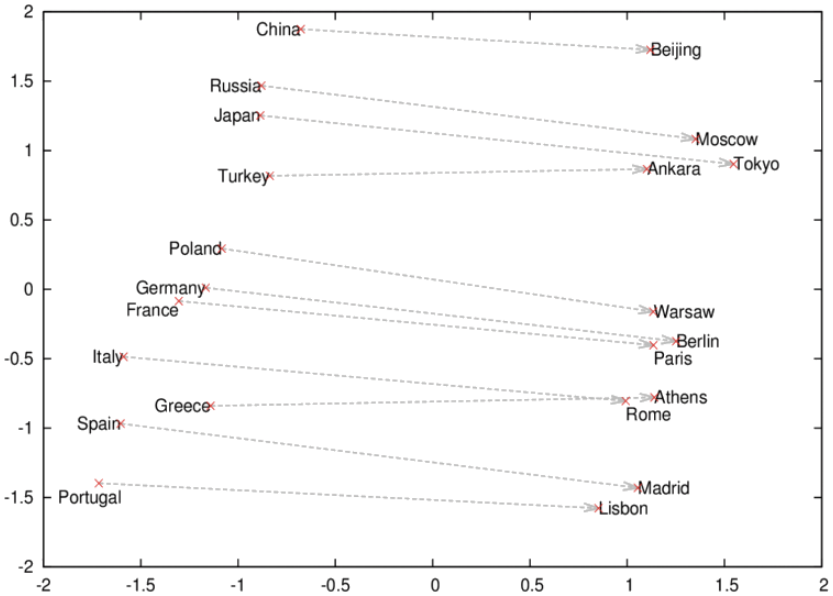



**Figure 6.1:** Two-dimensional PCA projection of the 1000-dimensional skip-gram vectors of countries and their capital cities [Mik13b]

Additionally, the resulting word embeddings capture linguistic regularities, for instance the vector operation $vec(\text{“President”}) - vec(\text{“Power”}) \approx vec(\text{“Prime Minister”})$. The semantic similarity between two words, which is important in the context of our work, denotes the cosine similarity between the words' Word2Vec vectors. The main goal in our work is to create entity embeddings instead of word embeddings and therefore we treat entity identifiers similar to words. Hence, after generating entity embeddings, the relatedness of two entities $e_j$ and $e_k$ is defined as the **cosine similarity** between the entities' vectors $vec(e_j)$ and $vec(e_k)$:

$$rel(e_j, e_k) = \frac{vec(e_j) \cdot vec(e_k)}{\|vec(e_j)\| \, \|vec(e_k)\|} \tag{6.3}$$

In contrast to other natural language, neural network models, the training speed of Word2Vec is very fast and can be further significantly improved by using parallel training. The training time on the Wikipedia corpus (without tables) using the Vector Space Modeling toolkit Gensim[1] [Řeh10] took $\approx 90$ minutes on our personal computer with a 4x3.4GHz Intel Core i7 processor and 16 GB RAM.

### 6.2.2 Corpus Generation

Word2Vec typically accepts a set of corpora containing natural language text as input and trains its word vectors according to the words' order in the corpora. Since we want to learn entity representations only, we have to create an appropriate Word2Vec input corpus file that exclusively comprises entities. The entities' order in the corpus file reflects how entities occur in entity-centric or document-centric KBs. In the following, we present how to create a suitable Word2Vec corpus basing on one or multiple KBs. By simply concatenating the algorithms' output files, we can combine the knowledge of both KB types.

#### *Entity-Centric Knowledge Bases*

Basically, the goal of language modeling (and thus of Word2Vec) is to estimate the likelihood of a specific sequence of words appearing in a (natural) language corpus. To this end, Word2Vec trains its word embeddings based on the order given in a document. In this work, we regard an entity-centric KB as a graph and present a generalization of language modeling by probing the graph via random walks. These walks can be thought of sentences or phrases in a special language. Learning entity embeddings based on these random walks allows us to extract two crucial information from the graph:

- The local link structure in form of the observed links between two entities.
- Latent relationships between two entities that are not directly connected but provide many common neighbors.

In the following, we assume that the entities defined in the entity-centric KB are directly connected via relations or connected via short relation paths.

1 http://radimrehurek.com/gensim/, last accessed on 2016-11-28

When we create a Word2Vec corpus, we produce a sequence of those resources that are in our entity target set $\Omega$. We regard an entity-centric KB as an **undirected** graph $G_{KB} = (V, E)$ where the nodes $V$ are the relevant entities in the KB, the edges $E$ denote relations between entities in the KBs and $x, y \in V, (x,y) \in E \Leftrightarrow \exists r : (x,r,y) \vee \exists r : (y,r,x)$ is a relation in the KB. After that, we perform a random walk on the graph $G_{KB}$. Whenever the random walk visits a node $x \in V$ we append the entity identifier of node $x$ to the output corpus file, if $x \in \Omega$. The succeeding node $succ(x)$ of $x$ is randomly selected by choosing an adjacent node of $x$, with probability $\frac{1}{edgesOf(x)}$. The function $edgesOf$ counts the number of edges that contact node $x$. We also introduce a random variable $X_x$ that provides jump probabilities to a specific node if a random jump is performed. A random jump can be seen as a new topic in a natural language text. We compute the jump probability from any node to a specific node $x$ by normalizing the inverse edge frequency $IEF$ of node $x$:

$$X_x = P(X_x = x) = \frac{IEF(x)}{\sum_{k \in V} IEF(k)} \tag{6.4}$$

$$IEF(x) = log(\frac{|E|}{edgesOf(x)}) \tag{6.5}$$

In our experiments, we used the parameter $\alpha = 0.1$ to perform a random jump. However, values of $0.05 < \alpha < 0.15$ practically do not affect the resulting Word2Vec model. Furthermore, the parameter $\theta$ specifies the number of random walks on the graph. We suggest to use $\theta = 5 * |E|$, which results in $\approx 50$ million random walks for DBpedia. Higher values of $\theta$ do not improve the entity embeddings but increase the training time. The corpus creation approach for RDF-KBs is explicated in Algorithm 1.

**Algorithm 1:** Creating a Word2Vec corpus based on entity-centric KBs

**input** : undirected graph G = (V, E), jump random variable $X_x$
**output** : Word2Vec corpus file
**parameter** : $\alpha$ node jump probability, $\theta$ number of walks

$corpus = createEmptyFile$
$x \longleftarrow drawRandomNode(X_x)$; $walks \longleftarrow 0$
**while** $walks < \theta$ **do**
  **if** $x \in \Omega$ **then**
    $appendToOutputFile(corpus, x)$
  **if** $randomInt(100) > (\alpha * 100)$ **then**
    $x \longleftarrow chooseNextNode(x)$ ; `// adjacent node:` $p(succ(x)) = \frac{1}{edgesOf(x)}$
  **else**
    $x \longleftarrow drawRandomNode(X_x)$
  $walks \longleftarrow walks + 1$;

***Document-Centric Knowledge Bases***

To create a Word2Vec corpus based on entity-annotated documents that contain natural language text, we assume to have the entity annotations in a unified format. Example 6.1 shows an example annotation in our document-centric format (cf. Section 5.2.2).

**Example 6.1.**
...English <e id="wiki:Computer_scientist">Computer Scientist</e>, logician...

Next, we iterate over all documents in the underlying corpus and replace all available, linked surface forms with its respective target entity identifier. The set of entity identifiers denotes the set of words in our special language. Further, all non-entity identifiers like words and punctuations are removed that all documents consist of entity identifiers separated by whitespaces only. However, the collocation of entities is still maintained as given by the original document. In this procedure, we aim to create an 'entity language', where the order of entities form a sentence or phrase. Similar to natural language text, successive entities are related since they describe the same topic. One might argue that this might be not the case in our entity language if the set of entity annotations in documents is sparse and the number of words removed between entities is large. However, we assume that if entity annotations are available in documents, then entities have been consistently annotated across the document in terms of quantity and uniformity. Nevertheless, the quality of the resulting entity embeddings depends on the number of entity annotations within the underlying documents. The resulting output documents of our algorithm are concatenated to create a single Word2Vec corpus file. The corpus creation approach for document-centric KBs is explicated in Algorithm 2.

**Algorithm 2:** Creating a Word2Vec corpus based on document-centric KBs

**input** : document corpus $C$
**output** : Word2Vec corpus file

```
1 corpus = createEmptyFile
2 forall D ∈ C do
3     D ⟵ replaceSFswithTargetIDs(D)
4     D ⟵ removeAllNonTargetIDs(D)
5     appendToOutputFile(corpus, D)
```

## 6.3 Approach

In this section, we present our approach to evaluate our entity relatedness measure. Basically, our system consists of the following three main steps: (i) index creation (Section 6.3.1), (ii) candidate generation (Section 6.3.2), and (iii) the assignment of entities to surface forms (Section 6.3.3). In the candidate generation step, we identify a set of possible candidate entities for each surface form and, thus, significantly reduce the number of possible target entities to improve performance and accuracy. To this end, we apply several heuristics proposed in [Usb14] or make use of known surface forms. In the final EL step, we use

this set of candidates to create a candidate entity graph. By applying the PageRank algorithm [Bri98; Whi03] we attempt to find the best possible entity configuration. More specifically, the candidate entity of a surface form that provides the highest PageRank score denotes the disambiguated target entity for that surface form. We use the PageRank algorithm because of its successful application in the EL task (e.g., [Alh14b; Han11b; Pic14]). We emphasize that the chosen algorithm can be seen as a baseline algorithm, which integrates a very simple local function and our entity relatedness measure for collective computation. As we shall see in the evaluation section later on, using this approach already leads to state-of-the-art results on most data sets since our robust relatedness measure outperforms existing techniques. In the following, we present each of the steps of our approach in more detail.

### 6.3.1 Index Creation

In our index creation process, we accept one or multiple source KBs that contain entity describing data. Basically, we accept entity-centric (e.g., DBpedia, YAGO3) and document-centric KBs (e.g., Wikipedia). In the next step, we use the given KBs to extract or compute three types of entity describing information and store them in an entity index:

- **Labels:** By default, we extract the entity names from the label attribute field within entity-centric KBs (e.g., *rdfs:label* in RDF-KBs) and store them in a label field. Further, in the case of document-centric KBs, we extract and store surface forms that have already been used to address a specific entity.
- **Semantic Entity Embeddings:** We store the respective vector for each entity in the embeddings field. In Section 6.2, we presented how to create these entity embeddings in detail.
- **Prior:** Generally, some entities occur more frequently than others. Thus, these popular entities provide a higher probability to re-occur in other documents. The Entity Prior $p(e_j)$ describes the a-priori probability that entity $e_j$ occurs (more details can be found in Section 3.2.2 on Page 32). We use the underlying KBs to compute Entity Priors by analyzing the number of its annotations in a document-centric KB or the number of in- and outgoing edges (e.g., relations between entities) within entity-centric KBs. In the former case, we use the normalized number of entity annotations across all annotated documents. In the latter case, we regard the KB as a directed graph, where the nodes $V$ denote entities, the edges $E$ are relations and $x$,$y \in V, (x, y) \in E \Leftrightarrow \exists r : (x, r, y)$ is a relation between two entities $x$ and $y$. Here, we use the number of in- and outgoing edges as quantity during the prior computations.

Given these information in an index, we link entities by selecting relevant candidates (Section 6.3.2) and computing the optimal entity assignments (Section 6.3.3).

### 6.3.2 Candidate Entity Generation

Given a constructed index, our approach accepts documents that contain one or multiple surface forms that should be linked to entities. In our EL chain, candidate entity generation

is the first crucial step. Our goal is to reduce the number of possible candidate entities for each input surface form by determining a set of relevant target entities. Hereby, we proceed as follows:

First, we compare the input surface form to those stored in the index. All entities in the index that provide an exact surface form matching serve as candidate entities.

Second, we use the candidate generation approach proposed by Usbeck et al. for AGDISTIS [Usb14] if no surface forms are available in the KBs (e.g., when using entity-centric KBs only). The authors suggested to apply a string normalization approach to the input text to eliminate plural and genitive forms, to remove common affixes such as postfixes for enterprise labels and to ignore candidates with time information within their label. Similar to AGDISTIS, our system compares the normalized surface forms to the labels in our index by applying trigram similarity. The trigram similarity threshold $\sigma = 0.82$ is constant in our system and experiments since it provides good results across all data sets and is the default setting in the AGDISTIS framework[1]. If an entity's label matches with the heuristically obtained label, while exceeding the trigram similarity threshold, and the entity is not yet a candidate for the surface form, the entity becomes a candidate.

### 6.3.3 Entity Linking Algorithm

After generating candidates for each surface form, we use the set of candidates to create a candidate entity graph. On this graph, we perform a random walk and determine the node relevance, which can be seen as the average number of its visits. The random walk is simulated by a PageRank algorithm that permits edge weights and non-uniformly-distributed random jumps [Bri98; Whi03].

First, we create a **complete, directed** $K$-partite graph whose set of nodes $V$ is divided in $K$ disjoint subsets $V_1$,...,$V_K$. $K$ refers to the number of surface forms $S$ and $V_i$ is the node set of generated candidates entities $\{e^i_1,...,e^i_{|V_i|}\}$ for surface form $m_i$. Since our graph is $K$-partite, there are only directed, weighted edges between candidate entities that belong to different surface forms. Connecting the entities that belong to the same surface form would be wrong since the correct target entities of surface forms are determined by the other surface forms' candidate entities (coherence).

The edge weights in our graph represent entity transition probabilities (ETP), which describe the likelihood to walk from a node (entity) to the adjacent node. We compute these probabilities by normalizing our entity relatedness measurement (cf. Equation 6.6). The relatedness between two entities is the cosine similarity ($cos$) of its entity embeddings (vectors) $vec(e^i_j)$ and $vec(e^h_k)$ stored in the index.

$$ETP(e^i_j, e^h_k) = \frac{cos(vec(e^i_j), vec(e^h_k))}{\sum_{l \in (V \setminus V_i)} cos(vec(e^i_j), vec(l))} \tag{6.6}$$

Given the current graph, we additionally integrate a possibility to jump from any node to any other node in the graph during the random walk with probability $\alpha = 0.1$. Typical

1 http://aksw.org/Projects/AGDISTIS.html, last accessed on 2016-11-28

values for alpha (according to the original paper [Whi03]) are in the range $[0.1, 0.2]$. We did not manually integrate jump edges in the graph (as in the transition case), but our PageRank algorithm simulates edges between all node pairs during PageRank computation. We compute a probability for each candidate entity being the next jump target. For this purpose, we use the already precomputed Entity Prior stored in our index.

Figure 6.2 shows a possible candidate entity graph when we have two surface forms 'TS' and 'New York' (cf. Example 3.1). The surface form 'TS' has only one candidate entity and consequently has already been linked to the entity *Time Square*. The second surface form 'New York' is still ambiguous, providing two candidate entities. We omit the jump probability values in this figure to improve visualization.

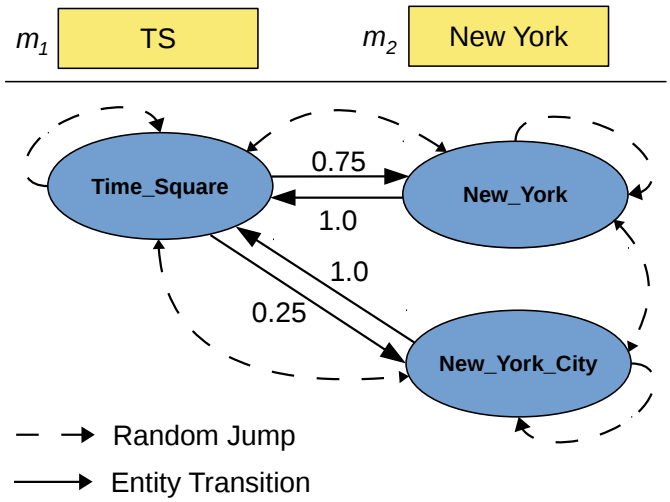



**Figure 6.2:** Candidate entity graph with candidates for the surface forms 'TS' and 'New York'. Solid lines denote entity transition probabilities and dashed lines denote jump probabilities between entity pairs.

After constructing the EL graph, we need to identify the correct candidate entity node. By applying the PageRank algorithm we compute a relevance score for each candidate entity. Afterwards, the candidate entity $e_j^i$ of a surface form candidate set $V_i$ that provides the highest relevance score is our entity result for surface form $m_i$.

## 6.4 Data Sets

In the following, we present seven well-known and publicly available data sets that are used in our evaluation. All data sets are integrated in the online EL evaluation framework GERBIL [Usb15] and strongly differ in document length and number of entities per document. Table 6.1 shows the statistics of our test corpora.

1. **ACE2004:** This data set from Ratinov et al. [Rat11] is a subset of the ACE2004 coreference documents and contains 57 news articles comprising 253 surface forms.
2. **AIDA-TestB:** The AIDA data set [Hof11] was derived from the CO-NLL 2003 task and contains 1393 news articles. The corpora was split into a training and two test corpora. The second test set has 231 documents with 19.40 entities on average.

**Table 6.1:** Statistics of our test data sets

| Data Set | Topic | #Doc. | #Ent. | Ent./Doc. | Annotation |
|---|---|---|---|---|---|
| ACE2004 | news | 57 | 253 | 4.44 | voting |
| AIDA-TestB | news/web | 231 | 4458 | 19.40 | voting |
| AQUAINT | news | 50 | 727 | 14.50 | voting |
| DBpedia Spotlight | news | 58 | 330 | 5.69 | domain experts |
| MSNBC | news | 20 | 658 | 32.90 | domain experts |
| N3-Reuters | news | 128 | 650 | 5.08 | voting |
| IITB | news/web | 103 | 11 245 | 109.01 | domain experts |
| Microposts | tweets | 1165 | 1440 | 1.24 | domain experts |
| N3-RSS 500 | RSS-feeds | 500 | 524 | 1.05 | domain experts |

3. **AQUAINT:** Compiled by Milne and Witten [Mil08b], the data set contains 50 documents and 727 surface forms from a news corpus from the Xinhua News Service, the New York Times and the Associated Press.
4. **DBpedia Spotlight:** The DBpedia Spotlight corpus was released together with its Spotlight system [Men11] and served as it's benchmark data set. It contains several non-named entities (e.g., entity *home*) within average-long textual paragraphs (i.e., few sentences). With 5.69 entities per document on average, this corpus provides enough entities for collective EL.
5. **MSNBC:** The corpus was presented by Cucerzan et al. [Cuc07] in 2007 and contains 20 news documents with 32.90 entities per document on average.
6. **N3-Reuters:** This corpus is based on the well-known Reuters-21578 corpus, which contains economic news articles. Roeder et al. proposed this corpus in [Röd14]. It contains 128 short documents with 4.85 entities on average.
7. **N3-RSS-500:** This corpus was published by Gerber et al. [Ger13] and is one of the N3 data sets [Röd14]. Originally, data was collected from 1457 RSS feeds, which included the data from all major worldwide newspapers across a wide range of topics (e.g., World, U.S., Business). A subset of this feed corpus was created by randomly selecting 1% of the contained sentences. Finally, domain experts annotated 500 sentences manually to create the N3-RSS-500 data set.
8. **IITB:** This manually created data set by Kulkarni et al. [Kul09] with 123 documents displays the highest entity/document-density of all data sets.
9. **Microposts-2014 Test:** The tweet data set was introduced for the 'Making Sense of Microposts' challenge and has very few entities per document on average [Usb15].

In GERBIL, there are a few more data sets that are not used in our evaluation. Those include two training data sets, which were used for supervised training purposes in other EL systems. Then, we have two additional AIDA test data sets that provide very similar queries as given by the AIDA-TestB data set. In terms of the KORE50 data set, we omit

the results due to its small number of entities (140 overall) and the resulting small number of entities per document.

## 6.5 Evaluation

In order to evaluate and analyze our new entity relatedness measure, we compare our approach to other state-of-the-art EL systems using entity-centric and document-centric KBs. Overall, the aim in our evaluation is four-fold. After discussing the experimental setup in Section 6.5.1, we first compare our approach to the current state-of-the-art named EL framework AGDISTIS [Usb14] that exclusively makes use of RDF data by default in Section 6.5.2. Second, we compare our Word2Vec embeddings to entity embeddings created with other state-of-the-art graph embedding approaches in Section 6.5.3. Third, in Section 6.5.4, we leverage the knowledge located in the document-centric KB Wikipedia and compare the results to the publicly available systems DBpedia Spotlight [Men11], AIDA [Hof11], Wikifier [Che13; Rat11], WAT [Pic14] and BabelFy [Mor14]. We emphasize that all approaches collectively link entities to surface forms, except for DBpedia Spotlight, which relies on non-collective features only. Finally, we investigate how our relatedness measure performs with erroneous information within entity-centric and document-centric KBs in Section 6.5.5.

### 6.5.1 Experimental Setup

Our approach is fully-implemented in Java and can be downloaded from our GitHub page[1]. To train our entity embeddings with Word2Vec, we chose Gensim [Řeh10], an open-source, robust and efficient framework to realize unsupervised semantic modeling from plain text. To evaluate our approach as well as the competitive EL systems, we use the D2KB task (i.e., EL task) in GERBIL v1.1.4 [Usb15]. The goal of the D2KB experimental type is to map a set of given surface forms to entities from a given KB. In our evaluation, we report the F1, recall and precision values aggregated across surface forms (micro-averaged). All values were automatically computed by GERBIL. For each section, we re-created the underlying EL index and, thus, re-trained all entity embeddings exclusively using the given KBs (i.e., either DBpedia, YAGO3 or Wikipedia).

In terms of parameters we distinguish between EL approach and Word2Vec parameters. Our proposed EL approach solely depends on one crucial parameter, the number of PageRank iterations. We empirically chose $it = 50$ iterations, which is the best trade-off between performance and accuracy in our experiments. When using Word2Vec, all embeddings were trained with $d = 400$ dimensions, which is suitable for millions of words (or entities) according to the original work [Mik13a]. Other Word2Vec parameters were also chosen as suggested in this work: skip-gram model, negative-sampling=10, window-size=10, minimum-count=1 and iterations=5.

### 6.5.2 Entity Linking Results on Entity-Centric Knowledge Bases

In our first evaluation, we analyze how our approach performs on entity-centric KBs. First, we compare our approach to AGDISTIS [Usb14], the current state-of-the-art named

1 http://github.com/quhfus/doser (ESWC branch), last accessed on 2016-11-28

EL framework for RDF-based KBs from 2014. Therefore, we use the current version of DBpedia (v.2015-10) since it is one of the most popular entity-centric KBs comprising general-domain entities. AGDISTIS performs best on this KB. We use the same entity target set as in AGDISTIS, consisting of named entities that belong to the persons, organizations or places class (cf. left column in Table 6.2). To the best of our knowledge, AGDISTIS is the only available approach that is able to perform named EL by using **only** DBpedia knowledge without implementation effort and significant accuracy drop. Further, we present the results after additionally considering the DBpedia category system during the training process of the entity embeddings (i.e., creating the Word2Vec corpus with including http://purl.org/dc/terms/subject). In this case, categories in DBpedia represent hidden nodes, which allow us to move along the DBpedia category graph during the random walk without storing the category nodes. Next, we investigate whether our approach performs better on the up-to-date YAGO3 KB, a KB originally derived from DBpedia and providing the same entities as available in DBpedia. Generally, AGDISTIS can be used with all kinds of RDF-KBs, but we were not able to configure the framework for the YAGO3 KB due to unresolvable error messages. Table 6.2 shows an overview of the entity classes used in our experiments.

**Table 6.2:** Class constraints for named entities (persons, organizations and places) only in DBpedia and YAGO3. Prefix **dbo** stands for http://dbpedia.org/ontology/, **foaf** for http://xmlns.com/foaf/0.1/ and **yago** for http://yago-knowledge.org/resource/.

| **Class** | **DBpedia** | **YAGO3** |
|---|---|---|
| Person | dbo:Person, foaf:Person | yago:yagoLegalActorGeo |
| Organization | dbo:Organization, dbo:WrittenWork | yago:yagoLegalActorGeo |
| Place | dbo:Place, yago:YagoGeoEntity | yago:yagoLegalActorGeo |

When comparing our approach to AGDISTIS on DBpedia (cf. Table 6.3), our system performs best on eight out of nine data sets, with and without using the DBpedia category system (denoted as *Our Approach* and *Our Approach - No Cat.*). Both variants attain similar results, but using the DBpedia categories further improves the F-measure by up to 3 percentage points. Despite applying the same candidate generation approach as proposed in AGDISTIS (because no external surface forms are available), our approach outperforms AGDISTIS by up to 10 F1 percentage points (IITB data set). On the other data sets (except MSNBC) the advantage is $\approx 5-6$ F1 percentage points. We assume that the ground truth entities in the MSNBC data set perfectly fit to available relations between entities in DBpedia. A look at the precision values shows that our approach links surface forms to entities more accurately (by up to 18% precision percentage points on Microposts-2014 Test). Overall, the bottle neck, which prevents achieving higher F-measures, is the absence of surface forms (resulting in a low recall) in the index.

Figure 6.3 shows the F1 values of our approach using DBpedia and YAGO3. We note that in this evaluation we regard both, relations and the respective category system of each KB. Nearly all results on YAGO3 are slightly worse than those attained on DBpedia ($\approx 2-3\%$ F1 percentage points). To analyze why DBpedia performs better than YAGO3, we use the labels extracted from DBpedia and the embeddings based on YAGO3. By

**Table 6.3:** F1, precision and recall values of our approach on 9 data sets using DBpedia

| Data set | Approach | F1 | Precision | Recall |
|---|---|---|---|---|
| ACE2004 | Our Approach | 0.702 | 0.795 | 0.629 |
| | Our Approach - No Cat. | **0.706** | **0.800** | **0.632** |
| | AGDISTIS | 0.658 | 0.696 | 0.624 |
| AIDA-TestB | Our Approach | **0.616** | **0.697** | **0.552** |
| | Our Approach - No Cat. | 0.602 | 0.684 | 0.537 |
| | AGDISTIS | 0.582 | 0.628 | 0.541 |
| AQUAINT | Our Approach | **0.646** | **0.820** | **0.533** |
| | Our Approach - No Cat. | 0.637 | 0.809 | 0.525 |
| | AGDISTIS | 0.596 | 0.739 | 0.499 |
| DBpedia Spotlight | Our Approach | 0.389 | 0.697 | 0.270 |
| | Our Approach - No Cat. | **0.403** | **0.710** | **0.281** |
| | AGDISTIS | 0.362 | 0.686 | 0.246 |
| MSNBC | Our Approach | 0.725 | 0.763 | 0.690 |
| | Our Approach - No Cat. | 0.727 | 0.765 | 0.692 |
| | AGDISTIS | **0.751** | **0.772** | **0.730** |
| N3-Reuters | Our Approach | **0.731** | **0.817** | **0.661** |
| | Our Approach - No Cat. | 0.713 | 0.791 | 0.649 |
| | AGDISTIS | 0.658 | 0.721 | 0.605 |
| N3 RSS-500 | Our Approach | 0.634 | 0.653 | 0.617 |
| | Our Approach - No Cat. | **0.648** | **0.667** | **0.630** |
| | AGDISTIS | 0.603 | 0.622 | 0.585 |
| IITB | Our Approach | **0.515** | **0.773** | **0.386** |
| | Our Approach - No Cat. | 0.488 | 0.751 | 0.362 |
| | AGDISTIS | 0.412 | 0.637 | 0.304 |
| Microposts-2014 Test | Our Approach | **0.489** | **0.763** | **0.360** |
| | Our Approach - No Cat. | 0.478 | 0.750 | 0.351 |
| | AGDISTIS | 0.428 | 0.584 | 0.337 |

obtaining the same results, we can say that the F1 difference between both KBs results from missing or wrong relations in the YAGO3 KB. However, we still outperform AGDISTIS on eight data sets by $\approx 4-5$ F1 percentage points.

In **summary**, our evaluation showed that our semantic embeddings perform better than binary relations even when there are no direct relations between specific entities in the KB. A significant drawback of binary relations is that two entities that are topically related but not directly connected via relation (i.e., edge distance $> 1$) are not captured. In contrast, our semantic embeddings are able to capture this topical relatedness. In terms of

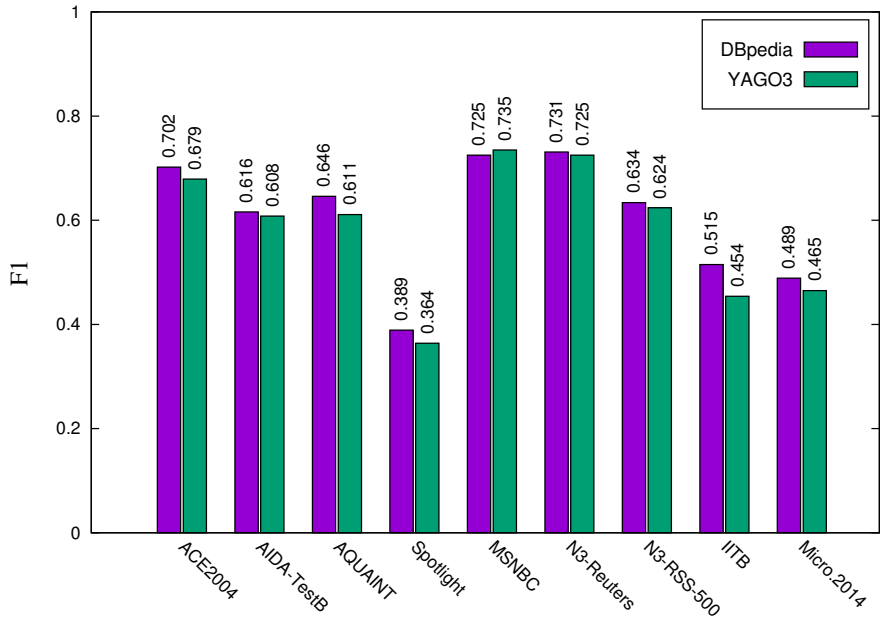



**Figure 6.3:** F1 values of our approach using DBpedia and YAGO3 as underlying KB. Our approach considers all relations and the category system of the respective KBs.

non-publicly available, graph-based entity-relatedness measures, the deep learning system by Huang et al. [Hua15] seems to perform best. Unfortunately, the authors exclusively evaluated their system on the AIDA CONLL-TestB data set as well as on a tweet data set.

### 6.5.3 Entity Linking Result with Other Graph Embedding Approaches

Recently, a couple of open source, network embedding algorithms were proposed in literature that are used for comparison to show the effectiveness of our approach. In order to compare our embeddings to those of other approaches in the context of EL, we downloaded the following three popular and state-of-the-art approaches for network embeddings and re-created the embeddings of all DBpedia entities:

- **DeepWalk:** The *DeepWalk* approach for social network embeddings by Perozzi et al. [Per14] is similar to our approach by using Word2Vec [Mik13a], more specifically, the skip-gram architecture. The authors suggested to perform multiple truncated random walks with each starting at the respective node (entity). Thus, this approach aims to create multiple contextual information fragments (similar to sentences in natural language text) by randomly navigating through different paths near the given node.
- **LINE:** *Line* by Tang et al. [Tan15] learns embeddings in a two-phase step. First, it learns $d/2$ dimensional embeddings by Breadth-First Search simulations over adjacent neighbors of a node to capture the local graph structure in form of the observed links. Second, it learns $d/2$ dimensional embeddings by randomly sampling nodes at a two-hop distance from the original node to capture latent relationships between

nodes (e.g., high similarity between unconnected nodes that provide many common neighbors).

- **Node2Vec:** The approach *Node2Vec* by Grover et al. [Gro16] defines a flexible notion of a node's neighborhood in the network. The authors achieved this by developing a family of random walks to explore the specific and varying neighborhoods of nodes across different networks. Overall, *Node2Vec* outperforms most other approaches on most networks given a specific optimization task (e.g., in the multi-label classification task).

For the sake of comparison, we exchanged our entity embeddings with those created with the respective algorithms and rerun all experiments. To provide fairness in the comparison, we trained the new embeddings with the same number of dimensions (i.e., $d = 400$). Further, we applied the default parameter settings during the training process. Table 6.4 shows the average F1 values of our approach after integrating the entity embeddings of our Word2Vec approach, DeepWalk, LINE and Node2Vec.

**Table 6.4:** Average F1 values across all data sets of Node2Vec, LINE, DeepWalk and our approach

| Our Approach | Node2Vec | LINE | DeepWalk |
|---|---|---|---|
| **0.600** | 0.598 | 0.592 | 0.595 |

In the original papers, significant accuracy differences were reported between the approaches (depending on the underlying data set and task). However, the differences remain marginal in our experiments. Although our approach achieves the highest F1 score across all data sets on average, the results of the other approaches are quite similar. The difference of the average F1 values across all data sets is $<= 0.8$ percentage points on all approaches. Despite the marginal differences in terms of EL accuracy, we emphasize that our approach is the fastest among all approaches in terms of training time (together with *DeepWalk*). Moreover, the underlying corpus creation algorithm is extremely simple and intuitive while proving excellent results.

### 6.5.4 Entity Linking Results on Document-Centric Knowledge Bases

In the following, we evaluate how our entity relatedness measure performs on document-centric KBs. To this end, we use Wikipedia (v.2016-06) as underlying KB. We compare the results of our approach to those achieved by the well-known EL systems Wikifier, DBpedia Spotlight, AIDA, WAT and Babelfy, which all leverage Wikipedia knowledge. DBpedia Spotlight, WAT and Babelfy are integrated in GERBIL by default, whereas we manually downloaded Wikifier and AIDA and installed them on our server with their best settings (i.e., 'Full Gurobi Configuration' for Wikifier and 'CocktailParty Configuration' for AIDA). Overall, the mentioned approaches return either entity identifiers for Wikipedia, DBpedia or YAGO. Since entities within these three KBs provide *sameAs* relations, we can easily compare the disambiguation accuracy while using the same data sets. Analyzing the competitor systems, to the best of our knowledge Wikifier is the current publicly available, state-of-the-art system for linking surface forms to Wikipedia pages regarding the average

accuracy across several data sets. We present the results of our approach when using DBpedia only (denoted as *O.A.*) and when using Wikipedia only (denoted as *O.A.Wiki*). We note that in this experiment, we let our approach link all entities in DBpedia (all entities belonging to the *owl:thing* class) instead of named entities only. Further, we make use of the DBpedia category system.

Table 6.5 shows the precision, recall and F1 values of our approaches using different data sources compared to other EL systems. We also provide the average F1 values across all data sets. Overall, the results of our approach *O.A.* are slightly worse than those of the previous experiment (cf. Section 6.5.2). This is because our index does not only contain named entities, and thus, the entity target set $\Omega$ comprises more entities to be disambiguated.

Using Wikipedia as KB in our approach (*O.A.Wiki*) significantly increases the average F1 values by nearly 15 F1 percentage points on average and significantly outperforms the other approaches. Our approach also outperforms the current state-of-the-art approach Wikifier on five out of nine data sets (ACE2004, MSNBC, N3-Reuters, N3-RSS-500 and Microposts2014-Test). In this context, it is noticeable that our precision and recall values are quite balanced compared to the other approaches. This can be explained, that other approaches likely use a more restricted candidate entity generation system. This leads to (much) lower recall values but more accurate results. Considering the AIDA-TestB data set, our approach performs comparatively poor with 72.2 F1 percentage points compared to 84.3 F1 percentage points by the WAT system. Analyzing the results on this data set shows that an analysis of the surface forms' textual context is necessary to perform better. For instance, given a set of location names as surface forms, our approach is not able to decide whether the surface forms refer to locations or football clubs. In contrast, on the ACE2004 and MSNBC data sets, our approach performs exceptionally well with 86.4 F1 and 88.1 F1 percentage points respectively.

A combination of DBpedia and Wikipedia embeddings does not lead to further improvements. We assume that the knowledge of DBpedia in form of relations is also integrated in Wikipedia and captured by our embeddings. Furthermore, we evaluated our approach with significantly less entity annotations in Wikipedia. Therefore, we randomly removed 80% of all Wikipedia entity annotations and re-trained our entity embeddings. We also re-computed the Sense Prior probabilities accordingly. With reduced training data our approach still achieves $\approx$ 69.4 F1 percentage points averaged across all data sets. Further reducing the number of entity annotations (i.e., omitting 90% Wikipedia annotations) leads to an average of $\approx$ 66.0 F1 percentage points across all data sets. With this experiment, we show that our entity relatedness measure also provides consistent results with a significantly reduced number of entity data in form of entity annotations.

In **summary**, we showed that knowledge in form of entity annotated documents is optimally captured by our entity embeddings. Our collective algorithm outperforms all other evaluated, publicly available, state-of-the-art approaches on several data sets when using Wikipedia as underlying KB. In contrast to other approaches, we did not consider the surrounding contextual words of the surface forms during EL.

**Table 6.5:** Micro-averaged precision, recall and F1 values of our approach on DBpedia, our approach on Wikipedia, Spotlight, Babelfy, AIDA, WAT and Wikifier on nine data sets

**Precision**

| Data set | O.A. | O.A. Wiki | Wikifier | Spot-light | AIDA | WAT | Babelfy |
|---|---|---|---|---|---|---|---|
| ACE2004 | 0.772 | **0.891** | 0.824 | **0.891** | 0.850 | 0.846 | 0.694 |
| AIDA-TestB | 0.680 | 0.723 | 0.777 | 0.789 | 0.775 | **0.852** | 0.809 |
| AQUAINT | 0.816 | 0.829 | **0.862** | 0.803 | 0.571 | 0.808 | 0.773 |
| DBpedia Spot. | 0.709 | 0.783 | 0.797 | **0.820** | - | 0.686 | 0.583 |
| IITB | 0.747 | 0.716 | **0.767** | 0.568 | 0.287 | 0.647 | 0.653 |
| Micro.2014 | **0.770** | 0.668 | 0.576 | 0.665 | 0.514 | 0.662 | 0.640 |
| MSNBC | 0.750 | 0.885 | **0.892** | 0.709 | 0.800 | 0.824 | 0.804 |
| N3-Reuters | **0.774** | **0.774** | 0.703 | 0.658 | 0.679 | 0.734 | 0.685 |
| N3-RSS-500 | 0.640 | 0.755 | 0.732 | 0.590 | 0.743 | 0.711 | **0.770** |

**Recall**

| Data set | O.A. | O.A. Wiki | Wikifier | Spot-light | AIDA | WAT | Babelfy |
|---|---|---|---|---|---|---|---|
| ACE2004 | 0.609 | **0.838** | 0.824 | 0.457 | 0.783 | 0.759 | 0.611 |
| AIDA-TestB | 0.532 | 0.721 | 0.777 | 0.518 | 0.774 | **0.836** | 0.794 |
| AQUAINT | 0.524 | 0.795 | **0.862** | 0.433 | 0.499 | 0.732 | 0.682 |
| DBpedia Spot. | 0.664 | 0.767 | **0.797** | 0.621 | - | 0.621 | 0.470 |
| IITB | 0.372 | 0.710 | **0.763** | 0.443 | 0.256 | 0.579 | 0.514 |
| Micro.2014 | 0.337 | **0.613** | 0.576 | 0.395 | 0.405 | 0.542 | 0.385 |
| MSNBC | 0.690 | **0.862** | 0.814 | 0.348 | 0.765 | 0.735 | 0.756 |
| N3-Reuters | 0.639 | 0.684 | **0.704** | 0.327 | 0.531 | 0.573 | 0.502 |
| N3-RSS-500 | 0.607 | 0.726 | **0.732** | 0.245 | 0.689 | 0.655 | 0.653 |

**F1**

| Data set | O.A. | O.A. Wiki | Wikifier | Spot-light | AIDA | WAT | Babelfy |
|---|---|---|---|---|---|---|---|
| ACE2004 | 0.681 | **0.864** | 0.824 | 0.605 | 0.815 | 0.800 | 0.650 |
| AIDA-TestB | 0.597 | 0.722 | 0.777 | 0.626 | 0.774 | **0.843** | 0.802 |
| AQUAINT | 0.638 | 0.820 | **0.862** | 0.563 | 0.533 | 0.768 | 0.725 |
| DBpedia Spot. | 0.686 | 0.775 | **0.797** | 0.707 | - | 0.652 | 0.520 |
| IITB | 0.497 | 0.713 | **0.765** | 0.497 | 0.270 | 0.611 | 0.576 |
| Micro.2014 | 0.469 | **0.639** | 0.576 | 0.495 | 0.453 | 0.595 | 0.480 |
| MSNBC | 0.719 | **0.881** | 0.851 | 0.467 | 0.782 | 0.777 | 0.779 |
| N3-Reuters | 0.700 | **0.727** | 0.694 | 0.436 | 0.596 | 0.644 | 0.579 |
| N3-RSS-500 | 0.623 | **0.740** | 0.732 | 0.346 | 0.716 | 0.682 | 0.707 |
| **Average** | 0.623 | **0.765** | 0.764 | 0.526 | 0.617 | 0.708 | 0.646 |

### 6.5.5 Noisy Knowledge Base Data

After presenting the results of our approach, we now investigate the robustness of our entity relatedness measure against noisy KB data. To this end, we conducted an experiment where we artificially changed annotations in our document-centric KB (i.e., Wikipedia) and relations in our entity-centric KB (i.e., DBpedia). We replaced the Sense Prior probabilities with a uniform jump distribution across all entities in our approach to exclusively use our entity relatedness measure to link the candidate entities (cf. Section 6.3.3). In the literature, it has been shown that the Sense Prior represents an influential EL feature, which might distort the results in this experiment. After this adaption, two main factors were still influencing the EL outcomes: (i) the candidate generation approach, and (ii) the quality of our entity embeddings. If the candidate generation approach generates only a single candidate entity for a surface form, then the entity embeddings are not decisive since the candidate entity clearly denotes the linked entity. As a consequence, we reduced the number of surface forms that provide a single candidate entity in our entity-centric approach. More specifically, we extracted all Wikipedia surface forms and used them during our evaluation on DBpedia **and** Wikipedia.

In terms of document-centric KBs, we performed similar to the approach proposed in Section 5.5.5 on Page 97. We iterated over all annotations in the corpus and selected an annotation to be wrong with probability $\alpha$. If we changed an annotation, we did not randomly exchange the target entity. Instead, we selected one of the wrong candidate entities in the sorted PageRank score list. To this end, we conducted the same experiments with our default approach (as evaluated before) and randomly chose a wrong candidate entity. We modeled this event with a Gaussian distributed random variable $X_{m_i} \sim \mathcal{N}(1, |LPR_{m_i}|)$. Variable $LPR_{m_i}$ denotes the descending-sorted entity list for surface form $m_i$ computed by our PageRank algorithm without the highest ranked candidate entity (i.e., the entity that is returned by our algorithm is omitted). Our random variable $X_{m_i}$ yields positive values only since negative values are useless in terms of selecting an entity position in the result list. Finally, we exchanged the original correct annotation with the allegedly wrong entity selected by the random variable. Algorithm 3 shows the process of adding noise to document-centric KBs, as done on Wikipedia.

**Algorithm 3:** Noise generator for document-centric KB

**input** : Document-centric KB $C$
**output** : Modified KB
**parameter** : $\alpha$ modification probability

**forall** $D \in C$ **do**
  **forall** $m_i \in D$ **do**
    **if** $randomInt(100) < (\alpha * 100)$ **then**
      $LPR_{m_i} = applyELAlgorithm(D)$
      $X_{m_i} \sim \mathcal{N}(1, |LPR_{m_i}|)$
      $assignTargetEntity(m_i, drawAndSelectEntity(X_{m_i}))$

When we conducted the experiment on the entity-centric KB DBpedia, we treated the KB as an undirected graph $G = (V, E)$. Further, we randomly iterated over all entities and then iterated over all existing relations (i.e., edges) of each entity. For each relation, we randomly modified the target with probability $\alpha$. To replace the target of a relation, we simply randomly chose an arbitrary entity. Finally, we exchanged the correct relation target entity with the new but wrong entity. Our noise generation algorithm for entity-centric KBs is explicated in Algorithm 4.

**Algorithm 4:** Noise generator for entity-centric KBs

**input** : Entity-centric KB as undirected Graph $G = (V, E)$
**output** : Modified KB
**parameter** : $\alpha$ modification probability

```
1 forall e_j ∈ V do
2 |   E_{e_j} = getEdgesOf(e_j)
3 |   forall edge ∈ E_{e_j} do
4 |   |   if randomInt(100) < (α * 100) then
5 |   |   |   edge.target = getRandomNode()
```

Figure 6.4 shows the average F1 values across all 9 data sets from 0% additional noise (as given by the original KBs) to 100% noise on Wikipedia and DBpedia. The initial F1 values achieved without noise are 0.729 on Wikipedia and 0.674 on DBpedia, respectively. The lower average F1 value on Wikipedia in comparison to those reported in Section 6.5.4 results from omitting the Sense Prior probability when computing the random jump probabilities. Further, an average F1 value of 0.274 with 100% noise in both approaches results from surface forms with only one (correctly) generated candidate entity. In this case, wrong entity embeddings do not have any influence on the outcome. Anyway, the results of both approaches are very similar with both approaches providing robust results up to 25% noise ($\approx -3$ percentage points F1). Further increasing the noise level up to 50% results in a moderate decrease of the average F1 values by up to $\approx 12$ F1 percentage points. We emphasize that a noise level of 50% in KBs is rather theoretical. For instance, even current (state-of-the-art) EL systems annotate documents quite accurately, i.e. $> 0.7$ F1, as shown in Section 6.5.4. Further increasing the noise level results in significantly decreasing F1 values since the entity embeddings tend to be randomly generated and, thus, lead to wrong results.

In **summary**, we showed that our entity relatedness measure provides Consistency in terms of poor quality entity definitions. More specifically, a noise rate lower than 50% led to slightly decreasing average F1 values with entity-centric and document-centric KBs. Further increasing the noise level led to a significant decrease in both approaches. Nevertheless, our experiments revealed that noisy data in form of automatically created entity annotations might be used as training data for entity embeddings to further include knowledge from external sources.

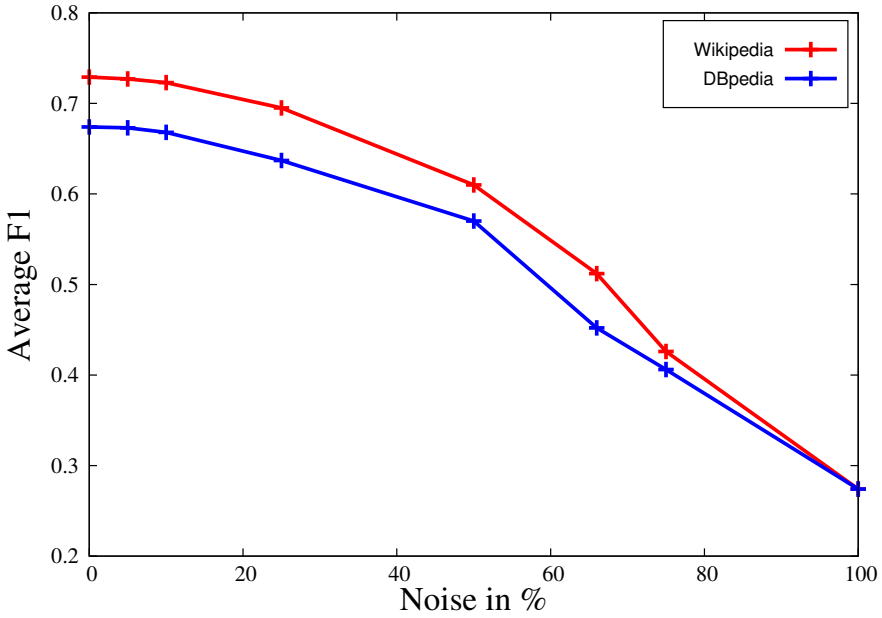



**Figure 6.4:** Influence of noisy data in Wikipedia and DBpedia on our entity embeddings

## 6.6 Conclusion

In this chapter, we presented a new, KB-agnostic, state-of-the-art entity relatedness measure based on semantic entity embeddings created with Word2Vec for collective EL. In this context, we proposed how to easily generate Word2Vec input corpora for entity-centric and document-centric KBs. To evaluate our relatedness measure, we integrated it in a graph-based, baseline EL approach that relies on the PageRank algorithm. In our evaluation, we showed that our approach outperforms other publicly available, state-of-the-art EL approaches that either rely on entity-centric (i.e., DBpedia) or document-centric KBs (i.e., Wikipedia). The competitor systems were Wikifier, DBpedia Spotlight, WAT, AIDA and Babelfy. Further, we investigated how our entity relatedness measure performs after increasing erroneous annotations or relations in the respective KBs. We showed that our created embeddings are robust against a moderate amount of noise (up to 50% noise). This implies that (textual) data annotated by automatic entity annotation systems can be used to further train our entity embeddings to integrate additional knowledge.

Overall, our measure provides Structural Robustness since it is KB-agnostic in terms of entity-centric and document-centric KBs. It also provides state-of-the-art results on different document types. We also assume that our measure performs well on tables since table cells within the same column are typically (highly) related. Our entity relatedness measure provides Consistency in terms of poor quality entity definitions, for instance in form of wrong relations in entity-centric KBs or wrong annotations in document-centric KBs. We also assume that our approach provides consistent results on any type of domain since it does not exploit any domain-dependent features. Finally, a first experiment showed that our measure performs well with few entity annotations in a document-centric KB. Indeed, we still have to confirm the results in form of in-depth experiments in the future.

# CHAPTER 7

## Textual Context

In this chapter, we present Doc2Vec as an effective technique to compare the surrounding textual context of surface forms to entity descriptions in Entity Linking (EL) systems. Overall, Doc2Vec provides Structural Robustness and consistent results with short entity descriptions. To evaluate the effectiveness of Doc2Vec, we compare it to other popular context matching techniques, such as the Vector Space Model (VSM) with TF-IDF weighted vectors, Okapi BM-25 and the probabilistic Entity-Context Model. Basically, we deploy a very simple, non-collective EL approach with context matching as its exclusive feature after candidate entity generation. In our evaluation, Doc2Vec provides the best robustness characteristics of all methods. It outperforms all other approaches on DBpedia and Wikipedia if the number of contextual surface form words is limited or only limited entity describing data is available in the respective knowledge base (KB).

The remainder of this chapter is structured as follows: In Section 7.1, we briefly introduce the chapter's core question, the methodology and the contributions. In Section 7.2, we propose Doc2Vec, an entity context matching technique based on semantic document embeddings. Section 7.3 describes our EL system and four often used textual context matching techniques, which serve as baseline techniques in our evaluation. In Section 7.4, we provide an in-depth evaluation of Doc2Vec on various data sets when using entity-centric and document-centric KBs (i.e., Wikipedia and DBpedia). Finally, we conclude the chapter in Section 7.5.

### 7.1 Introduction

Analyzing the surrounding context of surface forms to find the most relevant and correct candidate entity is the most intuitive EL approach. Hence, it is an important step in EL systems in practice. In natural language text documents, the surrounding surface form context typically denotes the words and phrases before and after a surface form. In contrast, in tables, the context of a surface form denotes the content of all table cells that are located in the same row as the respective surface form. However, the accuracy of context matching techniques also depends on the following two crucial factors:

- The type and structure of documents that contain surface forms due to differently long context passages.
- The quantity and quality of entity descriptions in the KB.

Most state-of-the-art EL approaches assume that enough entity describing data is available in KBs (cf. Section 3.2.4). These systems often neglect the importance of EL approaches

performing well with insufficient entity information (as is the case with special-domain entities) and/or short surface form contexts (as is the case with entities in tables or tweets).

In this chapter, we pose the following research question:

> **Research Question:** *Which context matching technique provides Structural Robustness and Consistency while achieving state-of-the-art results in EL systems?*

Similar to entity relatedness measures, a plethora of various context matching techniques has been proposed in the literature. In this chapter, we pick and investigate Doc2Vec as textual context matching technique in EL systems. It is an adaption of Word2Vec and achieves state-of-the-art results in *Semantic Textual Similarity* tasks [Lau16]. We compare Doc2Vec to other popular techniques in terms of their effectiveness in EL systems. Overall, we compare two TF-IDF-based approaches (i.e., VSM and Okapi BM-25), a language model approach, a Latent Dirichlet Allocation (LDA) approach and a neural network model approach (i.e., Doc2Vec). The other approaches were chosen according to their relevance in the literature and their implementation reproducibility. In contrast to Chapter 6, we directly integrated the respective approaches in an EL system. The approaches, or parts of them, are often implemented in publicly available, natural language processing toolkits, such as Apache Lucene[1], which simplifies the implementation.

Overall, our **contributions** in this chapter can be summarized as follows:

- We provide a systematic evaluation of Doc2Vec and four other state-of-the-art context matching techniques in EL approaches with regard to the quantity of entity descriptions, as provided by different KBs, and document structures and types.
- We show that Doc2Vec achieves state-of-the-art results while providing Structural Robustness and Consistency in terms of a low quantity of entity descriptions in the underlying KB.
- We show that the VSM with TF-IDF weighted vectors outperforms other state-of-the-art context matching approaches if a sufficient number of context words are given.

## 7.2 Textual Context Matching Based on Semantic Document Embeddings

How to capture the meaning of a document in a machine-understandable format is a central question of knowledge representation [Dai15]. The probably most established format is the bag-of-words model [Har54]. LDA [Ble03] is another very popular representation. As pointed out in Section 6.2, embeddings typically describe the representation of concepts by means of real numbers in a low-dimensional space. Recently, in addition to word embeddings [Ben03; Mik13a; Pen14], document-level embeddings have evolved to represent sentences, paragraphs or documents in a low-dimensional space. Popular representative works are [Kir15; Kus15; Le14]. In this chapter, we leverage Doc2Vec [Le14] to create entity-context embeddings to compute context similarity scores between the surrounding

1 http://lucene.apache.org/, last accessed on 2016-11-28

context of surface forms and entity descriptions. In Section 7.2.1, we introduce Doc2Vec and its different architectures. In Section 7.2.2, we briefly describe how we create our entity-context embeddings on the basis of entity-centric and document-centric KBs.

### 7.2.1 Doc2Vec

Doc2Vec is a modification of Word2Vec presented by Le and Mikolov [Le14]. It learns fixed-length embeddings from variable-length pieces of texts like documents. Throughout this chapter, we use the terms documents and paragraphs interchangeably. However, Doc2Vec addresses some of the key weaknesses of bag-of-word models by incorporating more semantics and considering the word order within a small context. As an example for the semantic embedding, the Doc2Vec model embeds the word 'powerful' closer to 'strong' than to 'Paris', which is not the case in bag-of-word models.

The architecture is either based on the distributed memory model (PV-DM), which is similar to the CBOW model of Word2Vec, or on the distributed bag-of-words model (PV-DBOW), which is similar to the skip-gram model. In the following, we describe both approaches in more detail.

***Distributed Memory Model***

The distributed memory model (PV-DM) is inspired by the continuous bag-of-words model in Word2Vec, which can be summarized as predicting a word given its context. While the word vectors being initialized randomly, they are adapted accordingly as a result of the prediction task during the training process. A very similar idea is used in the PV-DM model for Doc2Vec. In addition to word vectors, document or paragraph vectors contribute to the prediction of the next word given different contexts sampled from the respective paragraph. Figure 7.1 shows an example in the PV-DM model where a set of words and the respective paragraph id is used in the prediction task.

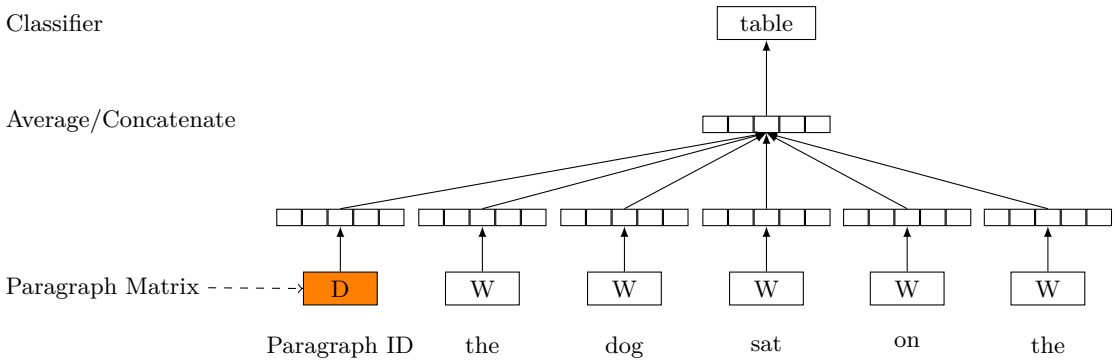



**Figure 7.1:** The distributed memory model is similar to the CBOW model of Word2Vec. An additional paragraph token is added to the context words and the concatenation or average of the paragraph vector with a context of multiple (five) words is used to predict the sixth word (inspired by [Le14]).

As depicted, in Doc2Vec, we make use of a document matrix $D$ (in addition to the word matrix $W$), which represents the paragraphs' weights (vectors). The document vector and

word vectors are averaged or concatenated to predict the next word in a context. The paragraph token can be thought of an additional word (vectors of documents and words are of equal size). It acts like a memory what is missing from the current context - or the topic of the paragraph. For this reason, this model is called the Distributed Memory Model of Doc2Vec [Le14]. The context size must be set a-priori and the context words are sampled from a sliding window over the given paragraph. The underlying paragraph vector is basically shared across all words within the paragraph but not across paragraphs. Further, the word matrix is shared across all available paragraphs in the corpus, e.g., the vector of a specific word is the same across all paragraphs.

A significant advantage of PV-DM is that it addresses the key weakness of bag-of-word models. While the word ordering in bag-of-word models is utterly ignored, PV-DM considers the word ordering in a small context, which is comparable to n-gram models with relatively large n. The authors claim that PV-DM might be better than bag of n-gram models since these models would create a very high-dimensional representation that tends to generalize poorly [Le14].

***Distributed Bag-of-Words Model***

While the distributed memory model concatenates/averages the word vectors and the paragraph vector to predict a context word, the distributed bag-of-words model works vice versa. It ignores the context words in the input and tries to predict words randomly sampled from the paragraph in the output. More technically, in each iteration the algorithm samples a text window, then samples a word within this text window and creates a classification task given the paragraph id (vector). We provide an example of the PV-DBOW model in Figure 7.2.

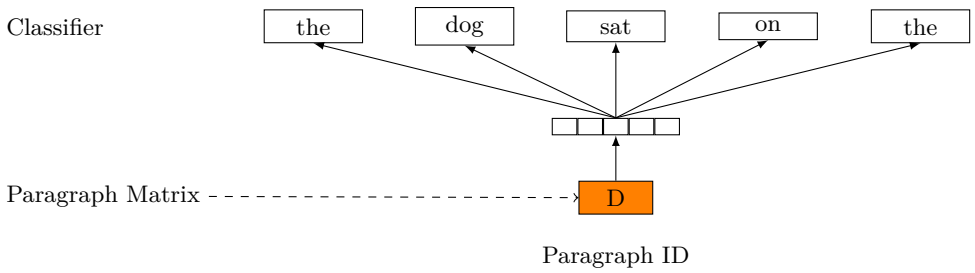



**Figure 7.2:** In the distributed bag-of-words model of Doc2Vec, the paragraph vector is trained to predict the words in a small window (inspired by [Le14]).

In comparison to PV-DM, this model is conceptually simpler and requires less memory during computation. Overall, we only need to store the softmax weights and the paragraph weights. In the previous model, we store the softmax weights, the word weights and the paragraph weights, respectively. We emphasize, that some frameworks that contain a PV-DBOW Doc2Vec implementation also offer to train word vectors to improve the paragraph vectors during the training process (e.g., Gensim [Řeh10]). However, this is out-of-scope in this work and we refer to the respective literature. The PV-DBOW model can be compared to the skip-gram model used in Word2Vec [Mik13a].

In **summary**, Doc2Vec provides different (significant) advantages over other baseline approaches:

- The word order is considered to preserve information in the paragraphs (PV-DM).
- Paragraph vectors inherit the semantics of words.
- Vectors are trained from unlabeled data, which becomes useful for tasks lacking enough labeled data.

Various authors were concerned about the optimal parameter tuning since the results of the original authors were hard to reproduce. For instance, Dai et al. [Dai15] provided a thorough comparison of Doc2Vec to other document modeling algorithms such as LDA on Wikipedia and arXiv. They also evaluated the accuracy of the method as they varied the dimensionality of the learned representations. Further, Lau and Baldwin [Lau16] provided a rigorous evaluation of Doc2Vec over the *Forum Question Duplication* and *Semantic Textual Similarity* task. In this work, the authors provided a significant number of parameter suggestions and evaluations for Wikipedia and other data sets. We used both papers to adapt our Doc2Vec parameter settings for our experiments.

Similar to Word2vec, we used the VSM toolkit Gensim[1] [Řeh10] to train our Doc2Vec model. The training time on the Wikipedia corpus took ≈ 2 days on our server with 20 cores and 25 GB RAM with 5 iterations overall.

### 7.2.2 Corpus Creation and Entity Context Matching Score

Before we are able to compute a context similarity score, we have to create an appropriate corpus comprising entity descriptions that is used to train the embeddings. Basically, any natural language source can be used that offers sufficient entity descriptions. A well-known example for entity describing documents are Wikipedia articles. Other entity-describing information like short descriptions in entity-centric KBs can also be leveraged (e.g., *dbo:abstract* in DBpedia). Finally, the training source needs to comprise a single document for each entity, or all documents need to be aggregated to a single corpus document. In this context, a simple way to create the Doc2Vec input corpus would be to let each line (labeled with the respective entity identifier) represent an entity description.

According to Lau and Baldwin [Lau16], this simple approach might lead to worse embeddings for longer documents due to significant length differences across all entity descriptions. Instead, the authors split longer entity descriptions into multiple sub documents. More specifically, they suggested to exploit paragraphs in documents and use them as single documents. Thus, we split Wikipedia articles into multiple paragraphs, leading to ≈ 9.3 million documents as Doc2Vec input in our experiments.

Another option to generate entity describing documents is concatenating the surrounding context of those surface forms that refer to the same entities across all annotated documents. Therefore, we extracted all surrounding contexts for an entity from a document-centric KB (i.e., Wikipedia) and created a surrounding context document by concatenating the context phrases.

1 http://radimrehurek.com/gensim/, last accessed on 2016-11-28

After generating the entity-context embeddings with Doc2Vec, we want to compute the similarity between an entity-context embedding and the surrounding context of a surface form. For that purpose, one needs to perform an inference step to compute the paragraph vector for the new surface form context. Basically, very similar to the training process, this is also obtained by gradient descent. In this step, the parameters of the already trained model, i.e., the word vectors $W$ and the softmax weights, are fixed.

When using a single document for describing an entity, we compute the cosine similarity between the inferred context vector and the entity-context vector. In contrast, if we subdivide each entity describing document into multiple paragraphs, we perform slightly different. We compute the cosine similarity between the inferred surrounding context vector of a surface form and the vector of each paragraph of the candidate entity. The highest cosine similarity is then returned as context matching score.

## 7.3 Approach

To evaluate how Doc2Vec performs in EL tasks, we compare the context matching approach to other popular approaches. More specifically, we compare Doc2Vec to the VSM with TF-IDF weights [Sal88], Okapi BM25 [Jon00], Entity-Context Model [Han11a] and the Thematic Context Distance with LDA model proposed in [Pil11] (all approaches are further explained below). To evaluate all techniques, we chose a simple yet very effective, non-collective EL approach consisting of the following three main steps: (i) index creation, (ii) candidate entity generation, and (iii) the assignment of entities to surface forms based on the respective textual context matching score.

During the **index creation** process, we create an EL index comprising two important information for each entity: (i) possible surface forms, and (ii) textual entity descriptions. Depending on the evaluated approach and underlying KB, we store different surface forms for the respective entities extracted from the original KBs (e.g., DBpedia, Wikipedia). Further, we store either plain entity descriptions, entity-context embeddings (when using Doc2Vec) or topic distributions (when using LDA).

In the **candidate entity generation** step, we aim to reduce the number of possible candidate entities for each input surface form by determining a set of relevant target entities $\Omega_i$ for each surface form $m_i$. To this end, we compare the input surface form $m_i$ to all labels stored in the index. All entities in the index that provide an exact surface form matching serve as candidate entities.

Finally, in the **candidate ranking** step, we rank the candidate entities according to their context matching relevance. More specifically, our EL algorithm computes a ranking $R_i$ of candidate entities given a surface form $m_i$, the corresponding surrounding context $c_i^\lambda$ and the set of candidate entities $\Omega_i$:

$$R_i = rank(m_i, c_i^\lambda, \Omega_i) \tag{7.1}$$

Parameter $\lambda$ denotes the number of words in front of and after a surface form. Given a ranked list for each surface form, our approach returns the highest ranked candidate entity as final assignment for a surface form.

For ranking purposes, we exclusively utilize the matching score of the respective context

matching technique. When using Doc2Vec, this is the cosine similarity between the inferred surrounding context vector and the entity-context embeddings of the candidate entities. In the following, we briefly describe popular context matching approaches for EL systems that are used in our evaluation for the sake of comparison. Generally, all approaches are flexible in terms of using entity descriptions as available in Wikipedia or entity context documents. The latter comprise the surface form contexts extracted from entity annotated documents for each entity. In our experiments, with used the entity descriptions by default.

***Vector Space Model with Lucene TF-IDF***

TF-IDF [Sal88], short for term frequency-inverse document frequency, is a numerical statistic that tries to reflect the importance of words in a document collection or corpus [Raj11]. It is typically used as a term weight in the VSM to compare documents via cosine similarity (cf. Equation 6.3). In the context of our approach, documents represent either bag-of-context-words $c_i^\lambda$ or the respective entity describing documents $d_j$ of our generated candidate entities $e_j \in \Omega_i$.

Generally, the term frequency (TF) $tf_{d_j}(w_k)$ denotes the number of occurrences of a specific term $w_k$ in an underlying (entity describing) document $d_j$. The inverse document frequency (IDF) $idf(w_k)$ weights term $w_k$ according to the number of occurrences across all documents: $idf_{w_k} = log\frac{|D|}{df_{w_k}}$, with $D$ denoting the (entity describing) document corpus and $df_{w_k}$ denoting the number of documents containing $w_k$ in $D$.

The high-performance, full-featured text search engine Apache Lucene[1] refines the default VSM score (cosine similarity of TF-IDF weighted vectors) for both search quality and usability[2]. In the following, we show Lucene's practical scoring function:

$$s(c_i^\lambda, d_j) = co(c_i^\lambda, d_j) \cdot qNorm(c_i^\lambda) \cdot \sum_{w_k \in c_i^\lambda} (tf_{d_j}(w_k) \cdot idf(w_k)^2 \cdot b(w_k) \cdot n(w_k, d_j)) \quad (7.2)$$

The function $co(c_i^\lambda, d_j)$ returns a score factor based on how many of the query terms in $c_i^\lambda$ are found in the entity describing document $d_j$. Documents that contain multiple query terms will receive a higher score than other documents with fewer query terms. Moreover, the function $qNorm(c_i^\lambda)$ is a normalizing factor used to make scores between queries comparable, but does not affect document ranking. The function $b(w_k)$ retrieves a boosting score for term $w_k$ in the query as specified in the query text. In our evaluation, however, we do not make use of any term boosting. Finally, function $n(w_k, d_j)$ integrates boost and length factors during indexing time: (i) Field Boost (not used during our Apache Lucene index construction), and (ii) lengthNorm - regards the number of the documents' tokens, with shorter documents contribute more to the score. Anyhow, when using the VSM with Lucene TF-IDF weights, we refer to Lucene TF-IDF throughout this chapter.

1 http://lucene.apache.org/, last accessed on 2016-11-28

2 http://lucene.apache.org/core/6_0_1/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html, last accessed on 2016-11-28

***Okapi BM-25***

The non-binary model, Okapi BM-25 weighting scheme, was developed as a way of building a probabilistic model sensitive to term frequency and document length [Jon00; Man08]. Similar to the VSM with TF-IDF weights, the BM-25 model is a bag-of-words retrieval function. It ranks the target documents (entity describing documents) according to the query terms (surface form context) appearing in each document. Using the same notation as before, the most typical form and also the version that is implemented in Apache Lucene, can be defined as follows:

$$s(c_i^{\lambda}, d_j) = \sum_{w_k \in c_i^{\lambda}} idf(w_k) \cdot \frac{tf_{d_j}(w_k)(\kappa + 1)}{tf_{d_j}(w_k) + \kappa(1 - \nu + \nu(\frac{L_{d_j}}{L_{avg}}))} \tag{7.3}$$

Here, $L_{d_j}$ denotes the length of the entity describing document $d_j$. Moreover, $L_{avg}$ denotes the average document length across the entire entity description corpus. Variable $\kappa$ is a positive tuning parameter to calibrate the document term frequency scaling. Further, $\nu$ is another tuning parameter, which determines the scaling by document length [Man08]. In terms of the $idf(w_k)$ function, BM-25 utilizes an alternative version:

$$idf(w_k) = log \frac{|D| - df_{w_k} + 0.5}{df_{w_k} + 0.5} \tag{7.4}$$

with $D$ denoting the entity description corpus and $df_{w_k}$ retrieving the number of documents containing term $w_k$.

***Entity-Context Model***

The Entity-Context Model was integrated by several EL systems (e.g., [Bar15; Gan16; Han11a]) and represents a multinomial unigram language model [Man08] (see Section 3.2.4). More specifically, we derive a specific language model $M_{d_j}$ for each entity describing document $d_j$. Then, we estimate the probability of generating the context query $c_i^{\lambda}$ according to each of these document models. More formally, this can be described as follows:

$$p(c_i^{\lambda}|M_{d_j}) = \prod_{w_k \in c_i^{\lambda}} p(w_k|M_{d_j}) \tag{7.5}$$

We estimate the respective probabilities $p(w_k|M_{d_j})$ as follows:

$$p(w_k|M_{d_j}) = \frac{count_{d_j}(w_k)}{\sum_{w_l \in T} count_{d_j}(w_l)} \tag{7.6}$$

The function $count_{d_j}(w_k)$ retrieves how often word $w_k$ has been annotated within the entity describing document $d_j$. A main problem is that a robust estimation of $p(w_k|M_{d_j})$ is often not possible due to the sparse data problem [Che96]. For that reason, we apply the Jelinek - Mercer smoothing method [Jel80] where we use the entire Wikipedia corpus as general language model. Similar to [Han11a], we use $\alpha = 0.2$ as smoothing parameter. As

the name describes, the Entity-Context Model is typically applied while utilizing existing entity contexts of the respective entities instead of using entity descriptions. Since the approach should also work with entity descriptions, we evaluate the approach with both, entity contexts and entity descriptions.

***Thematic Context Distance with LDA***

The Thematic Context Distance with LDA approach was proposed by Pilz and Paaß [Pil11]. It relies on the comparison of extracted topics from the surrounding context of surface forms and entity describing documents. Basically, we try to imitate the approach as proposed in [Pil11] in terms of algorithm and parameter settings. For a brief introduction to LDA, we refer to Section 3.2.4 in this work.

First, we train a LDA model based on a previously selected set of entity describing documents. This may be the set of Wikipedia articles, entity descriptions extracted from entity-centric KBs or documents comprising the concatenated contexts of surface forms that refer to the same entity. Anyway, for training purposes we use Gensim [Řeh10] with the same parameters as suggested here[1]. In terms of the best number of topics overall, we follow the suggestions in the original paper of $K = 200$. However, given the LDA model, for each word $w_k \in c_i^\lambda$ and $w_k \in d_j$, we infer the probability of belonging to topic $k$. Further, we derive the average probability of a topic $k$ describing context $c_i^\lambda$ or entity describing document $d_j$. This is done by averaging the probabilities of topic $k$ for each word $w_k$. As a result, we obtain a topic distribution for a surface form context and the description of a candidate entity. Finally, we compare the respective distributions by computing the Kullback-Leibler divergence [Kul51]. The Kullback-Leibler divergence achieved the best results for Thematic Context Distance computation in [Pil11]. The candidate entity with the lowest divergence value is selected as target entity by our ranking function.

## 7.4 Evaluation

To assess Doc2Vec in the contextual matching task within EL algorithms, we compare Doc2Vec to other popular context matching approaches that have often been used in the literature (cf. Section 7.3). After describing the experimental setup in Section 7.4.1, we evaluate our context matching techniques. First, we evaluate all approaches using the well-known document-centric KB Wikipedia in Section 7.4.2. Second, in Section 7.4.3, we present the results after re-conducting the experiments while using the entity descriptions located in the entity-centric KB DBpedia. Third, we present a parameter study and review how the Doc2Vec architectures PV-DM and PV-DBOW perform with various amounts of feature dimensions in Section 7.4.4.

### 7.4.1 Experimental Setup and Data Sets

All textual context matching techniques and our EL approach are fully implemented in Java and Python. For Lucene TF-IDF, Okapi BM-25 and the Entity-Context Model, we leverage the algorithms and features of the Apache Lucene 6.0.1 search engine. In order to create our Doc2Vec and LDA models, we utilized Gensim [Řeh10]. In our default

1 http://radimrehurek.com/gensim/wiki.html, last accessed on 2016-11-28

setup, we used all Wikipedia entity annotations (≈ 81 million annotations) to create a dictionary for candidate entity generation and stored them in our Apache Lucene index. Additionally, we stored all entity descriptions extracted from Wikipedia and DBpedia (version 2015-10) to facilitate our evaluation. When using DBpedia as entity describing KB, we utilize the surface forms from Wikipdia to generate candidate entities. Our entity index can be downloaded here[1]. In our evaluation, we report the results of the D2KB task (i.e., EL task) in GERBIL v1.1.4 [Usb15]. During evaluation, we report the F1 values aggregated across surface forms (micro-averaged). The recall and precision values are equal to the respective F1 values. This is because if no candidate entity can be found during the candidate generation process, we consider the entire entity target set as candidate entities. Thus, our approach definitely returns a target entity, which leads to equal F1, recall and precision values.

In terms of parameter settings, our approaches Lucene TF-IDF, Okapi BM-25 and Entity-Context Model use the default settings in Apache Lucene. All three approaches leverage the 'StandardAnalyzer' to preprocess the context and entity descriptions. Moreover, for the Entity-Context Model, we use the Jelinek - Mercer smoothing parameter $\alpha = 0.2$, which is adopted by the work of Han et al. [Han11a]. When we created our topic model with LDA, we used 200 topics overall as suggested by the authors of [Pil11]. For the rest, we applied the default settings as suggested by the Gensim developers. Regarding Doc2Vec, we first split all Wikipedia entity descriptions into multiple paragraphs resulting in ≈ 9.3 million documents that were used to create the Doc2Vec input corpus. During paragraph generation, we leveraged the information provided by the Wikipedia syntax. Selecting the optimal parameter settings for Doc2Vec was not an easy task. We considered the information published in the works by Zwicklbauer et al. [Zwi16b] and Lau and Baldwin [Lau16], who provided multiple experiments with Doc2Vec on Wikipedia. According to the conducted parameter settings in Section 7.4.4, we chose the Doc2Vec architecture PV-DM with $d = 400$ dimensions for each paragraph in our evaluation. Other Doc2Vec parameters were also chosen as suggested in the mentioned works: PV-DM, negative-sampling=5, window-size=5, minimum-count=8 and iterations=5.

In our evaluation, we use the same data sets as described in Chapter 6, namely ACE2004, AIDA-TestB, AQUAINT, DBpedia Spotlight, MSNBC, N3-Reuters, N3-RSS-500, IITB and Microposts-2014 Test. Additionally, we use the KORE50 data set [Hof12] in GERBIL, since it provides very short contexts. KORE50 contains 50 hand-crafted, difficult test sentences from the celebrities, music, business, sports and politics domain. Each test document contains 14 words per sentence on average. Finally, to complete our data sets, we also use the Web-Manual table data set which was crawled by Limaye et al. [Lim10]. The data set comprises a huge number of 51 898 cells, but only 9239 of them are annotated with ground truth entities. We build the surrounding surface form context in the underlying tables by concatenating all non-entity cells of the same surface form row. Table 7.1 shows an overview of our data sets with statistics being relevant in our evaluation. We classified all data sets according to the average context length into short, medium, long or table documents to

1 http://github.com/quhfus/DoSeR/wiki/Disambiguation-Index, last accessed on 2016-11-28

**Table 7.1:** Important data set statistics for our textual context matching experiments

| Data Set | #Surface Forms | ∅ Context Length | ∅ Candidates | Length Classification |
|---|---|---|---|---|
| DBpedia Spotlight | 330 | 33.9 | 60.1 | Short |
| KORE50 | 148 | 13.9 | 127.3 | Short |
| Microposts | 1440 | 18.7 | 41.9 | Short |
| N3-RSS 500 | 524 | 29.9 | 43.2 | Short |
| ACE2004 | 253 | 518.2 | 60.6 | Medium |
| AQUAINT | 727 | 275.9 | 28.9 | Medium |
| N3-Reuters | 650 | 235.3 | 58.2 | Medium |
| IITB | 11 245 | 775.7 | 25.5 | Long |
| MSNBC | 658 | 672.0 | 63.8 | Long |
| AIDA-TestB | 4458 | 228.9 | 56.8 | Medium/Table |
| Web-Manual | 9239 | 15.9 | 45.6 | Table |

simplify the discussion later. Further, we print the average number of candidate entities per surface form. The high number of candidates per surface form results from the bulk of surface form labels in our index. This increases the difficulty of disambiguating entities by exclusively analyzing the textual context.

### 7.4.2 Comparing Textual Context Matching Techniques on Wikipedia

In the following, we analyze how Doc2Vec performs in terms of textual context matching in EL algorithms on the document-centric KB Wikipedia. More specifically, we compare Doc2Vec to other popular textual context matching approaches used in the literature, namely: Lucene TF-IDF, Okapi BM-25, Entity-Context Model and Thematic Context Distance with LDA (cf. Section 7.3). In the respective experiments, we utilized the entities' Wikipedia articles as entity describing documents by default.

In this section, our evaluation is three-fold. First, we start with an analysis of how the context matching approaches perform with various context lengths (i.e., 20, 50, 100, 150, 200 and 600 words overall). A surface form context typically comprises a set of words before and after the surface form. For instance, a context length of 200 words denotes the context of 100 words before and after the respective surface form. In related work, most authors determined a fixed context-length for their approach in their experiments while omitting an in-depth context-length evaluation. Second, based on the optimal context-length for each approach, we analyze the EL accuracy on each data set separately. Third, we further prune our candidate entity set for each surface form as it is typically done in existing EL systems and discuss the results of our context matching techniques after this step.

***Context Length Evaluation***

In our first evaluation, we compare all 5 approaches given a specific number of context words and report the average F1 values across all data sets. In addition, we report the

results of a random entity assignment, which randomly selects a candidate entity to be the linked target entity. Figure 7.3 shows an overview of how the methods perform with various numbers of context words.

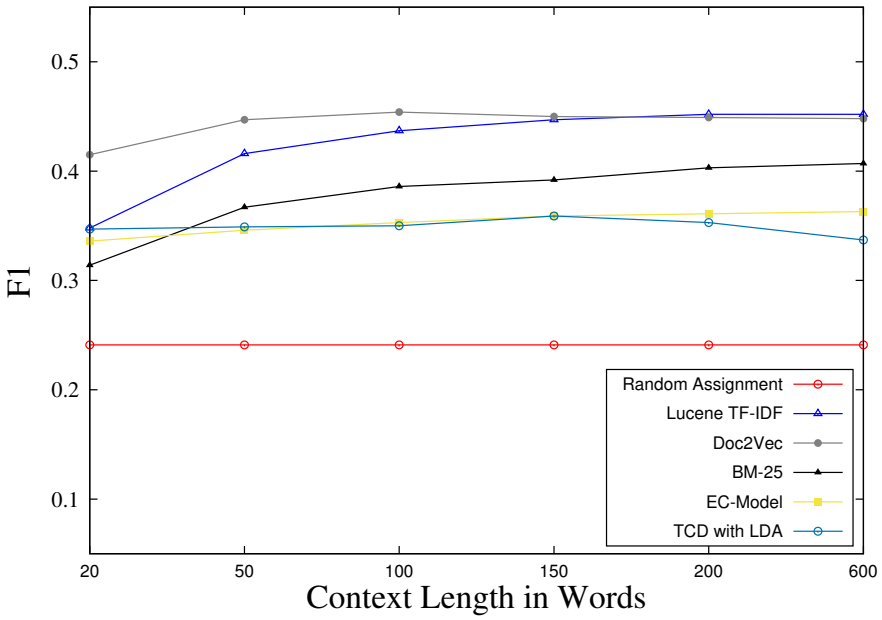



**Figure 7.3:** Micro-averaged F1 values of 5 textual context matching techniques and a random assignment with various context lengths across 11 data sets

Randomly assigning candidate entities marks a decent baseline with a F1 value of $\approx 0.24$. This is largely because a couple of surface forms do only provide a single (correct) candidate entity. Lucene TF-IDF and Doc2Vec perform best with F1 values of $\approx 0.45$ with 600 and 100 context words, respectively. Reducing the number of context words leads to a (significant) decrease of the average F1 value when using Lucene TF-IDF, while the results with Doc2Vec remain more constant. Hence, we assume that Doc2Vec superiorly captures the content of short context fragments which are typically exhibited in tweets and tables. When using additional context words, Lucene TF-IDF and Doc2Vec achieve similar results on average. However, the curve in Figure 7.3 of the Okapi BM-25 Apache Lucene implementation is similar to that of Lucene TF-IDF while constantly lacking $\approx 5$ F1 percentage points. Finally, the results of the Entity-Context Model and Thematic Context Distance with LDA are surprisingly poor although their main EL approaches in the original papers achieved comparatively strong results. In both techniques, more context words do not positively influence EL results, i.e., the results remain constant with an F1 value of $\approx 0.33$ with various context lengths.

***Data-Set-Specific Evaluation***

In the following, we further investigate and analyze the EL accuracy of our textual context matching techniques on each data set. To this end, we selected the best context length for each approach according to its best average F1 value across all data sets (cf. Figure 7.3). More specifically, we chose a context length of 100 words for Doc2Vec, a context length of

150 words for Thematic Context Distance with LDA and a maximum context length of 600 words for all other approaches. Table 7.2 shows the micro-averaged F1 values of our 5 context matching techniques and the random assignment approach on 11 data sets. The data sets are subdivided according to our data set classification in Section 7.4.1.

**Table 7.2:** Micro-averaged F1 values of Doc2Vec, Lucene TF-IDF, Okapi BM-25, Entity-Context Model, Thematic Context Distance and Random Assignment. The entity descriptions were extracted from **Wikipedia**.

| Data Set | Doc2Vec | Lucene TF-IDF | Okapi BM-25 | EC Model | TCD | Random Assign. |
|---|---|---|---|---|---|---|
| DBpedia Spot. | **0.374** | 0.345 | 0.306 | 0.291 | 0.296 | 0.202 |
| KORE50 | **0.338** | 0.273 | 0.224 | 0.106 | 0.096 | 0.020 |
| Micro. 2014 | **0.301** | 0.293 | 0.260 | 0.281 | 0.289 | 0.188 |
| N3 RSS-500 | 0.563 | **0.571** | 0.536 | 0.510 | 0.513 | 0.372 |
| ACE2004 | 0.421 | **0.482** | 0.395 | 0.359 | 0.357 | 0.259 |
| AQUAINT | 0.516 | **0.535** | 0.481 | 0.403 | 0.396 | 0.305 |
| N3-Reuters | 0.588 | **0.593** | 0.566 | 0.510 | 0.506 | 0.383 |
| IITB | **0.381** | 0.353 | 0.329 | 0.330 | 0.324 | 0.190 |
| MSNBC | 0.603 | **0.682** | 0.612 | 0.472 | 0.436 | 0.272 |
| AIDA-TestB | 0.423 | **0.459** | 0.418 | 0.316 | 0.317 | 0.214 |
| Web-Manual | **0.503** | 0.387 | 0.355 | 0.392 | 0.377 | 0.229 |

Overall, Lucene TF-IDF provides the best results on 6 out 11 data sets while Doc2Vec performs best on 5 data sets. After digging deeper into the results, we claim that Doc2Vec basically performs best on data sets with short and very short contextual phrases as it is the case in short and table data sets. Doc2Vec leads all other approaches on the short documents in the DBpedia Spotlight, KORE50, N3-RSS-500 and Microposts-2014 Test data sets. Additionally, it significantly outperforms the competitors on the Web-Manual table data set by at least $\approx 8$ F1 percentage points. On the AIDA-TestB data set, which contains table-like documents, Lucene TF-IDF outperforms Doc2Vec by $\approx 3$ F1 percentage points. Here, we assume that Lucene TF-IDF better captures the short document description at the beginning of each document due to a significantly longer context length in this experiment (600 words). When taking only a limited number of context words (i.e., Doc2Vec with 100 context words), Doc2Vec does not consider the important table headline which consequently leads to decreased F1 values on this data set. When we increase the context length to 600 words, Doc2Vec achieves a F1 value of 0.482 on the AIDA-TestB data set. However, the story looks different on long data sets. Regarding IITB and MSNBC, Lucene TF-IDF performs best by a significant margin ($\approx 3$ and 8 F1 percentage points). The margin becomes closer on the medium-length documents. On the AQUAINT and N3-Reuters-128 data sets, the differences between the Doc2Vec and Lucene TF-IDF F1 values remain marginal. All other approaches, in particular the Entity-Context Model and Topical Context Distance with LDA, perform poorly and even

get outperformed by the Okapi BM-25 approach.

The Wikipedia KB belongs to document-centric KBs. Other document-centric KBs typically do not comprise entity-describing documents as it is the case in Wikipedia. Thus, generalizing the achieved results in this section to all document-centric KBs is not possible. To this end, we also evaluate how Doc2Vec and Lucene TF-IDF perform when using the surrounding context of annotated entities as entity descriptions. Therefore, we concatenated the contextual phrases of those surface forms that refer to the same entity. For each surface form, we extracted 200 words before and after the surface form. In terms of Doc2Vec corpus generation, we concatenated five extracted contexts to form a paragraph with each representing a training document. Our evaluation shows that both, Doc2Vec and Lucene TF-IDF results decrease across all data sets. Doc2Vec achieves an average F1 value of 0.42 (−3.1 F1 percentage points) and TF-IDF achieves an average F1 value of 0.40 (−5.4 F1 percentage points) across all data sets. The results show that surface form context matching is also suitable to compute a matching score between an entity and a surface form. As explained in Section 7.3, the Entity-Context Model has been leveraged to compute a matching score between the current surface form context and the surface form context of previously annotated entities. Hence, we also conducted an experiment whether the approach performs better with contexts instead of entity descriptions. However, the average F1 value slightly increases from 0.355 to 0.370. In contrast to the other approaches, the result values of the Entity-Context Model slightly increase, but are still worse than those attained by the Doc2Vec and Lucene TF-IDF approach.

***Data-Set-Specific Evaluation with Pruning***

In our next evaluation, we investigate the results after additionally integrating a candidate pruning step into our approach. More specifically, we use the Sense Prior probability $p(e_j|m_i)$ to estimate the probability of seeing an entity with a given surface form. We select the top-10 entities as the candidates to keep the popular candidates. As shown by many other works, the Sense Prior is a very strong baseline algorithm and, hence, we assume that the correct ground truth entity is still kept in the pruned candidate entity list. Table 7.3 shows the micro-averaged F1 values on 11 data sets sorted according to our short, medium, long and table classification. Generally, the margin between the approaches achieved F1 values decreases while attaining much higher F1 scores on average. Regarding the high number of generated candidate entities for each surface form (cf. Table 7.1), the candidate list is now significantly shortened by our pruning approach. This leads to significantly better results due to less target candidates. Similar to the previous experiments, the Entity-Context Model and Thematic Context Distance with LDA perform poorly despite the number of candidate entities being reduced.

***Summary and Further Discussion***

In summary, we showed that Lucene TF-IDF and Doc2Vec perform equally well when comparing the surrounding context of surface forms with existing entity descriptions extracted from Wikipedia. However, each approach has its strengths and deficits. While Lucene TF-IDF performs best on documents with sufficiently long contexts, Doc2Vec significantly outperforms the other approaches on short and table documents. Further, it

**Table 7.3:** Micro-averaged F1 values of Doc2Vec, Lucene TF-IDF, Okapi BM-25, Entity-Context Model, Thematic Context Distance and Random Assignment after candidate entity pruning to 10 candidates. The entity descriptions were extracted from **Wikipedia**.

| Data Set | Doc2Vec | Lucene TF-IDF | Okapi BM-25 | EC Model | TCD | Random Assign. |
|---|---|---|---|---|---|---|
| DBpedia Spot. | **0.498** | 0.448 | 0.403 | 0.329 | 0.322 | 0.202 |
| KORE50 | **0.379** | 0.327 | 0.279 | 0.129 | 0.137 | 0.020 |
| Micro. 2014 | **0.385** | 0.382 | 0.337 | 0.345 | 0.327 | 0.188 |
| N3 RSS-500 | **0.598** | 0.592 | 0.558 | 0.511 | 0.531 | 0.372 |
| ACE2004 | 0.579 | **0.644** | 0.527 | 0.399 | 0.405 | 0.259 |
| AQUAINT | **0.616** | 0.604 | 0.557 | 0.441 | 0.460 | 0.305 |
| N3-Reuters | 0.642 | **0.668** | 0.601 | 0.510 | 0.512 | 0.383 |
| IITB | 0.429 | **0.462** | 0.431 | 0.348 | 0.345 | 0.190 |
| MSNBC | 0.657 | **0.693** | 0.627 | 0.460 | 0.482 | 0.272 |
| AIDA-TestB | 0.535 | **0.555** | 0.491 | 0.347 | 0.353 | 0.214 |
| Web-Manual | **0.550** | 0.474 | 0.464 | 0.506 | 0.463 | 0.188 |

is interesting to see that Doc2Vec achieves the best F1 values with a moderate number of context words, while all other approaches perform better after increasing the context size. We do not evaluate the approaches' performance because we have not integrated a method to quickly access the Doc2Vec embeddings for optimal performance.

### 7.4.3 Comparing Textual Context Matching Techniques on DBpedia

In our last section, we provided a general evaluation of 5 context matching techniques on Wikipedia. However, since Wikipedia is unique in terms of entity describing documents, we also evaluate the approaches on the corresponding entity-centric KB DBpedia. Existing entity-centric KBs like DBpedia, YAGO or Uniprot typically do not contain long entity descriptions as provided by Wikipedia. For robust EL, there is a need for context matching techniques that perform sufficiently well with short entity descriptions. The experiments conducted in this section are very similar to those of the previous section while generating and using context models based on DBpedia instead of Wikipedia. We also rely on the optimal context lengths settings for each approach as determined in Figure 7.3. We re-conducted the respective length experiment with DBpedia, but came to the same conclusions as before, where we use a context length of 100 words for Doc2Vec, 150 words for TCD with LDA and 600 words for the other approaches.

Table 7.4 shows the F1 values of five context matching approaches and a random assignment using entity descriptions extracted from DBpedia[1]. Additionally, we provide a direct comparison of Doc2Vec and Lucene TF-IDF using Wikipedia and DBpedia entity descriptions in Figure 7.4. Generally, the F1 values of all approaches on all data sets have

1 Extracted from http://dbpedia.org/ontology/abstract, last accessed on 2016-11-28

**Table 7.4:** Micro-averaged F1 values of Doc2Vec, Lucene TF-IDF, Okapi BM-25, Entity-Context Model, Thematic Context Distance and Random Assignment when using **DBpedia** entity description

| Data Set | Doc2Vec | Lucene TF-IDF | Okapi BM-25 | EC Model | TCD | Random Assign. |
|---|---|---|---|---|---|---|
| DBpedia Spot. | **0.340** | 0.232 | 0.241 | 0.253 | 0.263 | 0.202 |
| KORE50 | **0.295** | 0.143 | 0.143 | 0.097 | 0.105 | 0.020 |
| Micro. 2014 | **0.283** | 0.165 | 0.158 | 0.175 | 0.160 | 0.188 |
| N3 RSS-500 | **0.480** | 0.351 | 0.343 | 0.337 | 0.313 | 0.372 |
| ACE2004 | **0.355** | 0.351 | 0.327 | 0.312 | 0.314 | 0.259 |
| AQUAINT | 0.424 | **0.444** | **0.444** | 0.397 | 0.386 | 0.305 |
| N3-Reuters | **0.524** | 0.462 | 0.462 | 0.449 | 0.442 | 0.383 |
| IITB | 0.339 | **0.344** | 0.333 | 0.340 | 0.295 | 0.190 |
| MSNBC | 0.416 | 0.446 | **0.452** | 0.385 | 0.363 | 0.272 |
| AIDA-TestB | **0.337** | 0.306 | 0.290 | 0.266 | 0.255 | 0.214 |
| Web-Manual | **0.384** | 0.286 | 0.299 | 0.291 | 0.308 | 0.188 |

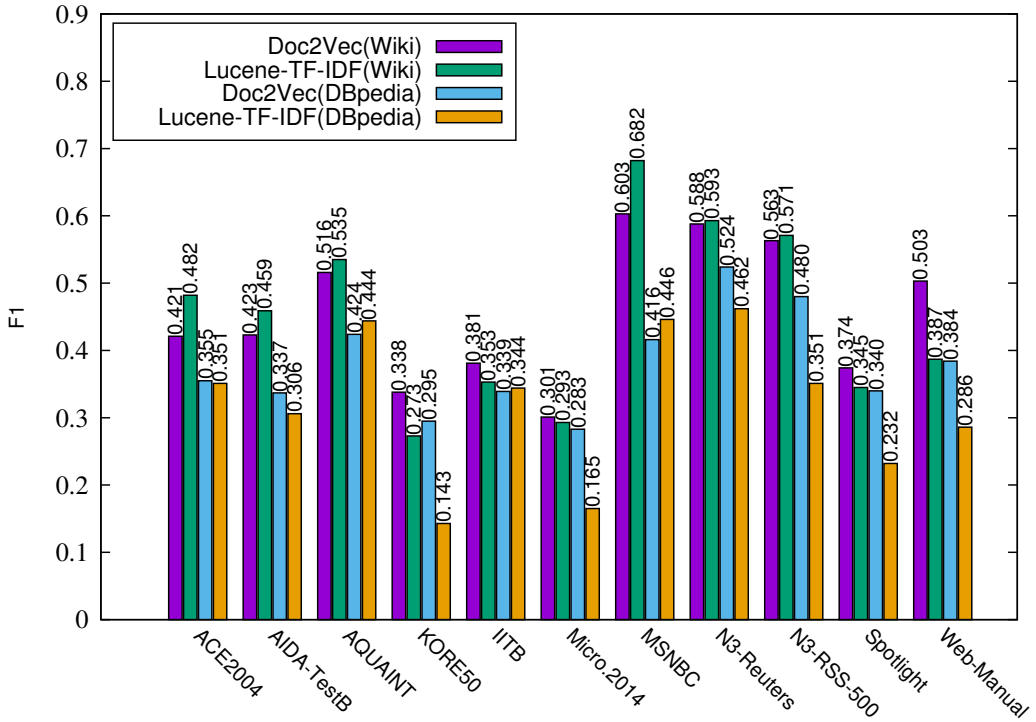



**Figure 7.4:** Micro-averaged F1 values of Doc2Vec and Lucene TF-IDF when using Wikipedia and DBpedia entity descriptions on 11 data sets

decreased due to less extensive entity descriptions. In the Wikipedia experiments, the average F1 values between Doc2Vec and Lucene TF-IDF were nearly the same. Now, the gap between the Doc2Vec and Lucene TF-IDF average F1 values has increased to $\approx 6$ F1 percentage points across all data sets in favor of Doc2Vec (0.379 and 0.321 average F1 values). It is important to note the significant margin on the short and table data sets. Regarding the long data sets like IITB and MSNBC, Lucene TF-IDF still outperforms Doc2Vec by $\approx 4$ F1 percentage points. Conducting additional experiments with other surface form context lengths led to similar results to those achieved in Section 7.4.2: Again, Doc2Vec is not able to take advantage of additional context words. In contrast, approaches like Lucene TF-IDF and Okapi BM-25 are not able to capture the semantics of short contexts to provide better results. Similar to the Wikipedia experiments, the Entity Context Model and the Textual Context Distance with LDA perform poorly on DBpedia. In contrast, Okapi BM-25 achieves surprisingly strong results on the AQUAINT and MSNBC data sets and also achieves very similar results to Lucene TF-IDF on all data sets when using DBpedia.

We **summarize** that the F1 values of all evaluated approaches decrease on DBpedia due to its short entity descriptions. However, Doc2Vec outperforms all competitors by a significant margin on nearly all data sets when using DBpedia as underlying KB. The margin of the F1 values between Doc2Vec and Lucene TF-IDF increases on data sets with short surface form contexts as it is typically the case in table or tweet documents. Taking the results of the previous section into account, we emphasize that Doc2Vec provides Structural Robustness since it performs consistently well on nearly all types of data sets (i.e., table, short, medium and long documents) and also on entity-centric and document-centric KBs.

### 7.4.4 Doc2Vec Parameter Study

The Doc2Vec implementation in Gensim provides a wealth of parameter settings, which may influence the resulting document embeddings. In our experiments, we chose the settings that were suggested in the works [Dai15; Lau16; Zwi16b] (negative-sampling=5, window-size=5, minimum-count=8 and iterations=200). In the following, we compare the PV-DM and PV-DBOW Doc2Vec models with a various number of dimensions to determine the best settings for the EL task.

Figure 7.5 depicts the micro-averaged F1 values across all data sets of our EL approach when using either the PV-DM or PV-DBOW Doc2Vec architecture and a specific number of dimensions. The printed results refer to Doc2Vec models based on the Wikipedia paragraph corpus whose construction was explained earlier. Both architectures achieve the best results with $d = 400$ dimensions, with PV-DM leading PV-DBOW by $\approx 2$ F1 percentage points. One reason for this outcome might be, that in contrast to PV-DBOW, PV-DM takes the word order into consideration, at least in a small context, in the same way that an n-gram model with a large n would do [Le14]. It is also very interesting to see that the averaged F1 values of both architectures with 800 dimensions drop by up to 5 F1 percentage points compared to $d = 400$. One reason might be that a high number of dimensions leads to some kind of overfitting and, thus, the optimal number of dimensions for embeddings probably depends on the number of entities and amount of training data [Zwi16b]. Similar to the

results achieved in [Zwi16b], PV-DBOW provides slightly more robust F1 results with less dimensions. With 200 or less dimensions, PV-DBOW tops its counterpart by up to $\approx 3$ F1 percentage points. Since we are interested in the best overall results, we suggest to use PV-DM for context matching in EL systems in the future. Anyhow, a careful analysis of the underlying corpus and an adaption to the Doc2Vec parameter settings is required to affirm the results.

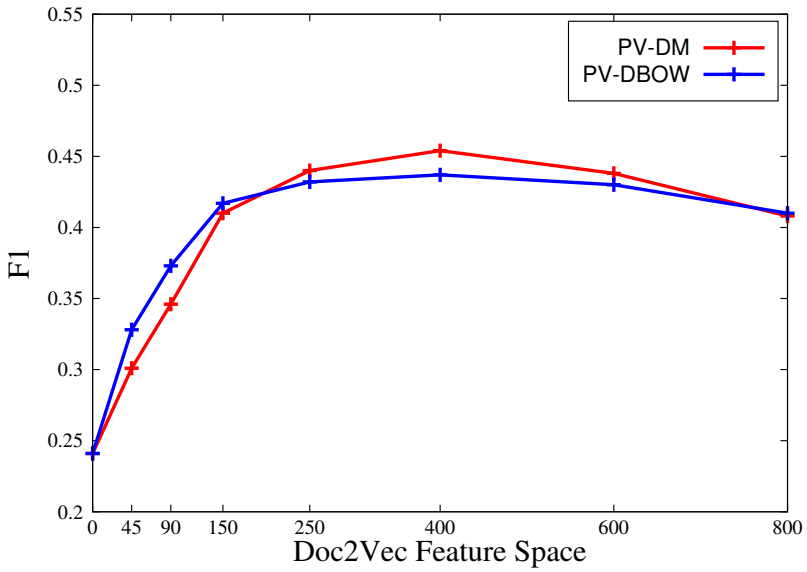



**Figure 7.5:** Micro-averaged F1 values of our approach with the Doc2Vec architectures PV-DM and PV-DBOW and a various number of dimensions

In **summary**, we note that in our experiments with Wikipedia paragraphs as documents, the PV-DM architecture performs best when using a specific number of dimensions. We note that further increasing the number of dimensions leads to a (significant) decrease of EL results with both architectures. However, the optimal dimension number has to be re-determined for other KBs.

## 7.5 Conclusion

In this chapter, we presented the neural-network-based approach Doc2Vec as a textual context matching technique for robust EL. In this context, we provided a systematic comparison to four other popular context matching techniques. These are the VSM with Apache Lucene TF-IDF weights, Okapi BM-25, the Entity-Context Model and the Thematic Context Distance with LDA. In our experiments, we evaluated all approaches by first determining the respectively best textual context length across all data sets measured in words. Then, we analyzed and discussed the results of all approaches on different data sets when using Wikipedia as entity describing source. Further, we re-conducted the data set experiments and used entity descriptions located in the entity-centric KB DBpedia to investigate the robustness of the approaches in terms of short entity descriptions. Finally, we provided a parameter study of the Doc2Vec architectures PV-DM and PV-DBOW.

Overall, our results revealed that our context matching technique based on Doc2Vec

achieves state-of-the-art results on most data sets and provides Structural Robustness. Moreover, it provides consistent results with (very) short entity descriptions in the underlying KB. We also showed that the VSM with adapted TF-IDF weights outperforms other state-of-the-art context matching techniques if a sufficient number of surface form context words is given.

A **limitation** of our work is the small number of evaluated KBs. With Wikipedia and DBpedia denoting general-domain KBs, we cannot entirely generalize the achieved results to any (special-domain) KB. Moreover, we did not evaluate the performance of the context matching techniques. We are going to tackle both types of experiments in the near future.

# Part IV

## A Robust Entity Linking System

# CHAPTER 8

## DoSeR - Disambiguation of Semantic Resources

In this chapter, we combine the results of the previous chapter to construct DoSeR (**D**isambiguation **o**f **Se**mantic **R**esources), a robust (i.e., providing Structural Robustness and Consistency), state-of-the-art Entity Linking (EL) system. DoSeR is a knowledge base (KB) agnostic EL framework that extracts relevant entity information from multiple (entity-centric and document-centric) KBs in a fully automatic way. The main EL algorithm in DoSeR utilizes semantic entity and document embeddings for entity relatedness and textual context matching computation and represents a new collective, graph-based approach. Our approach is also able to abstain if no appropriate entity can be found for a specific surface form. In our evaluation, we analyze how DoSeR performs on general-domain KBs (i.e., Wikipedia, DBpedia, YAGO3) and special-domain KBs (e.g., Uniprot). We compare DoSeR to other publicly (e.g., Wikifier [Rat11]) and non-publicly (e.g., Probabilistic Bag-Of- Hyperlinks model [Gan16]) available EL systems. Our system achieves significantly (>5%) better results than all other publicly available approaches on various document structures and types (e.g., news, tables). This chapter partially covers the ideas, findings and materials published in the works [Zwi16a] and [Zwi16b].

The remainder of this chapter is structured as follows: After introducing the chapter in Section 8.1, we provide an overview of the DoSeR framework in Section 8.2. Section 8.3 presents the data sets used in our evaluation. In Section 8.4, we describe the experimental setup and the achieved results. We conclude the chapter in Section 8.5.

### 8.1 Introduction

The ultimate goal and main research question in this work is to create a robust EL system in terms of Structural Robustness and Consistency. To this end, we first analyzed three crucial components of EL algorithms to gain new insights into techniques and algorithms whose usage essentially influence Robustness in EL systems. These components are the underlying KB, the entity relatedness measure and the textual context matching technique. Overall, we revealed the following three core findings in terms of robust EL, which we aim to consider in our EL framework:

1. **Knowledge Bases (Chapter 5):** We showed that a federated approach leveraging knowledge from entity-centric and document-centric KBs can (significantly) improve the Consistency of EL systems.
2. **Entity Relatedness (Chapter 6):** We proposed a new state-of-the-art entity relatedness measure that provides Structural Robustness and consistent results with a low quantity and poor quality of entity definitions.

3. **Textual Context (Chapter 7):** We presented Doc2Vec as textual context matching technique that provides Structural Robustness and consistent results with long and short entity definitions.

Based on these findings, we present DoSeR, a robust EL framework in terms of Structural Robustness and Consistency that achieves state-of-the-art results on various KBs, domains, and document structures and types. DoSeR is KB-agnostic in order to complement entity-centric and document-centric KBs in terms of entity coverage, i.e., the total number of entities available in a KB, and entity description, i.e., the completeness and quality of the description of one entity. Further, the graph-based EL algorithm in DoSeR unifies our proposed and robust semantic entity embeddings (cf. Chapter 6) for collective EL and entity-context embeddings (cf. Chapter 7) for surrounding context matching. In the case of our algorithm being uncertain about the correct entity target, our approach abstains by returning the pseudo-entity *NIL*.

In particular, we provide the following **contributions**:

- We present DoSeR, a new state-of-the-art (named) EL framework that emphasizes Robustness in terms of Structural Robustness and Consistency.
- We evaluate our algorithm against other state-of-the-art EL systems on 16 data sets overall and show that our approach outperforms all other systems by a significant margin on nearly all data sets.
- We discuss the influence of the quality of the underlying KB on the EL accuracy and indicate that our algorithm achieves better results than non-publicly available state-of-the-art algorithms.
- We provide our EL system as well as the underlying KB as open source solutions[1]. These resources allow a fair comparison between future EL algorithms and our approach that are not biased by the KB.

## 8.2 DoSeR Framework

In the following, we present the DoSeR framework, which consists of two major parts: Preprocessing and EL algorithm. First of all, we provide a brief overview of the entire EL system in Section 8.2.1. Then, in Section 8.2.2, we describe our index construction process, which extracts and stores entity data from various KBs. In Section 8.2.3, we explain our main EL algorithm in detail. Finally, we describe the important subparts of our EL algorithm, candidate entity generation and graph generation, in Section 8.2.4 and 8.2.5.

### 8.2.1 Overview

In this section, we present the architecture of the DoSeR framework (cf. Figure 8.1). DoSeR accepts entity-centric and/or document-centric KBs as input and consists of the following three main steps: (i) index creation (Section 8.2.2), (ii) candidate entity generation (Section 8.2.4), and (iii) the assignment of entities to surface forms (Section 8.2.3).

1 http://github.com/quhfus/DoSeR/, last accessed on 2016-11-28

The first step in the index creation process is to define a set of *core* KBs. The set of core KBs (depicted with a continuous line in Figure 8.1) is used to specify the set of target entities $\Omega$ that should be linked by our framework. In the following, DoSeR processes the contents of all given (core and optional) KBs and stores available surface forms from document-centric KBs, entity embeddings as well as a-prior probability for each entity (optional KBs are figured with a dashed line in Figure 8.1). This KB preprocessing step is executed only the first time or if the data of a new KB should be integrated. After preprocessing, DoSeR accepts documents with surface forms (e.g., manually marked by users) that should be linked to entities.

In the candidate generation step, we identify a set of possible candidate entities for each surface form and, thus, significantly reduce the number of possible target entities. In the main EL step, we first further reduce the number of candidates by means of a semantic candidate filter. Then, we use the set of candidates to create a candidate entity graph. By applying a two-phase personalized PageRank algorithm, we attempt to find the best possible entity configuration. In the following, we present each of the steps of DoSeR in more detail.

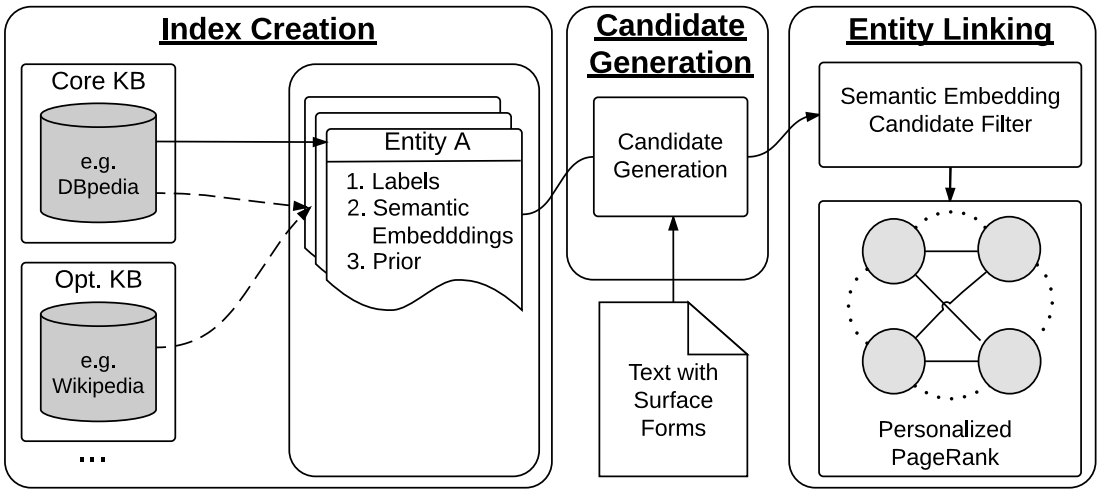



**Figure 8.1:** Overview of the DoSeR framework

### 8.2.2 Index Creation

Before starting the index creation process, we first have to choose one or multiple source KBs that contain entity describing data. Basically, DoSeR accepts entity-centric KBs (e.g., DBpedia, YAGO3) and document-centric KBs (e.g., Wikipedia). Since document-centric KBs do not have a standardized format, DoSeR requires a unified format for annotated entities. Our Example 6.1 on Page 106 shows an example annotation of the format that DoSeR uses internally.

Next, given a set of source KBs in the appropriate format, we have to select a set of *core* KBs. The set of all entities specified or annotated in these core KBs specifies our target entity set $\Omega$. If the core KBs provide information about the entities' classes (e.g., *rdf:type*), $\Omega$ can be restricted to named entities only (e.g., persons, organizations and places). After

specifying $\Omega$, all core and optional KBs are used as data sources for our target entity set $\Omega$. Optional KBs complement the core KBs in terms of completeness and quantity/quality of entity descriptions. Overall, our approach is fully KB-agnostic in terms of entity-centric and document-centric KBs. In the next step, DoSeR creates an index comprising the following three entity describing information:

- **Labels:** By default, DoSeR extracts the *rdfs:label* attribute of all given entity-centric KBs and stores them in a label field. Our approach can be configured to use any set of properties as label. Further, DoSeR searches for document-centric KBs in our specified KB set and, if available, extracts and stores surface forms that have been used to address a specific entity.
- **Semantic Embeddings:** DoSeR automatically creates semantic embeddings for all entities regardless of the underlying KBs. We distinguish between **entity embeddings** and **entity-context embeddings**. The former are created with Word2Vec to compute an entity relatedness score between an entity pair (as proposed in Chapter 6). The latter are created with Doc2Vec to compute a matching how good this entity fits to the context of a surface form (as proposed in Chapter 7). The Word2Vec model is trained based on the knowledge of all available entity-centric and document-centric KBs as explained in the respective chapter. The Doc2Vec model is trained on an entity-description corpus, which comprises entity describing information extracted and combined across one or multiple KBs. For instance, when using Wikipedia and DBpedia, we leverage the entities' article text of Wikipedia and the entities' abstract text of DBpedia (i.e., *dbo: abstract* ). However, our approach also allows us to utilize the surrounding contexts of entity annotations in document-centric KBs.
- **Prior Probabilities:** The prior probability describes how likely an entity occurs (together with a specific surface form) within a document. Depending on the underlying KBs, we compute and store either the Entity Prior $p(e_j)$ or the Sense Prior $p(e_j|m_i)$. The prior probability is less meaningful and is exclusively computed if no document-centric KBs are available. In this case, we regard the entity-centric KB(s) as a directed graph, where the nodes $V$ denote entities, the edges $E$ are relations and $x$,$y \in V, (x, y) \in E \Leftrightarrow \exists r : (x, r, y)$ is a relation between two entities $x$ and $y$. Here, we use the number of in- and outgoing edges as quantity during the prior computations (cf. Section 6.3.1). If document-centric KBs are available, we use the number of entity annotations with specific surface forms in these documents to compute the Sense Prior. The computation of the prior probabilities is explained in Section 3.2.2 on Page 32.

Given these information in an index, we are able to apply our EL algorithm to collectively link the entities defined in the core KBs.

### 8.2.3 DoSeR Entity Linking Algorithm

Given the previously constructed EL index, our algorithm accepts documents that contain one or multiple surface forms that should be linked to entities. It links all surface forms within a document using a collective, graph-based approach. Overall, given a set of surface

forms, our algorithm tries to seek the optimal entity assignment $\Gamma$ and can be subdivided into four main steps. Algorithm 5 gives an overview of the entire process, whose steps are explained in the following.

***Candidate Entity Generation***
The first step in our EL chain is *Candidate Entity Generation*. The goal is to reduce the number of possible candidate entities for each input surface form $m_i$ by determining a set of relevant target entities, the target candidate entity set $\Omega_i$ for surface form $m_i$. Details of our candidate generation process are described in Section 8.2.4. Given the candidates we link surface forms with none or one candidate entity. We also initialize the entity set $E_d$ with the entities of unambiguous surface forms or already linked surface forms (Lines 2-7).

***Semantic Embedding Candidate Filter***
Our second step *Semantic Embedding Candidate Filter* filters candidate entities that fit to the general topic described by the already disambiguated entities (Lines 8-17) requiring at least 3 already assigned entities. The underlying assumption is, that all entities in a paragraph are somehow topically related. To infer this general topic, we create a topic vector $tv = \sum_{e_j \in E_d} vec(e_j)$, with $E_d$ being the set of already linked entities and $vec(e_j)$ being the entity embedding of entity $e_j$ (Word2Vec vector). Next, we compute the semantic similarity (cosine similarity) between the general topic vector $tv$ and the candidate entities of all not yet disambiguated surface forms. If the similarity exceeds the a-priori given *CandidateFilter* threshold $\lambda$, the candidate entity remains in the candidate list of the respective surface form. If no candidate of a specific surface form exceeds the threshold, the candidate set for this surface forms remains unchanged. We note that this filter is a crucial step toward fast and accurate EL. Omitting this step results in a significantly lower performance combined with decreasing results ($\approx$ 2 to 5 percentage points F1, depending on the data set).

***High Probability Candidate Linking***
The third step *High Probability Candidate Linking* comprises the PageRank application on an EL graph to link high probability candidates (Lines 18-24). Detailed information for graph construction and PageRank can be found in Section 8.2.5. Next, we rank the candidate entities for each surface form according to their relevance score given by the PageRank algorithm in descending order. Additionally, we select the highest PageRank score $h$, second-highest PageRank score $s$ and average PageRank score $avg$ across all entities that belong to the same surface form. Given these parameters, we define a threshold $dynThreshold$ for determining the certainty in the ranking based on the differences between the first and the second ranked candidate:

$$dynThreshold = h - margin_1 \cdot (h - avg) \tag{8.1}$$

whereas details on the parameter $margin_1$ are discussed in Section 8.4.1. We use this threshold as a certainty criterion, indicating whether the top-ranked candidate entity of a surface form is the correct target. More specifically, if the PageRank score $s$ of the second

**Algorithm 5:** Our graph-based EL algorithm integrated in DoSeR

**input** : $M =< m_1,...,m_S >$ ,Threshold $\lambda$, $margin_1$, $margin_2$

**output** : Assignment $\Gamma =< t_j^1,...,t_k^S >$, with $t_j^i$ denoting the assigned entity $e_j$ of $m_i$

configuration $\Gamma = tuple()$; linked entities $E_d = \emptyset$; candidate set $\Omega_i = \emptyset$

`// Candidate Entity Generation`

**for** $m_i \in M$ **do**
  $\Omega_i =$ generateCandidates$(m_i)$
  **if** $|\Omega_i| = 0$ **then**
    $\Gamma(i) = NIL$
  **else if** $|\Omega_i| = 1$ **then**
    $\Gamma(i) = e_j \in \Omega_i$; $E_d = E_d \cup \Omega_i$

`// Semantic Embedding Candidate Filter`

**if** $|E_d| > 2$ **then**
  **for** $m_i \in M$ *and* $|\Omega_i| > 1$ **do**
    $set = \emptyset$
    **for** $e_j \in \Omega_i$ **do**
      **if** $cosineSim(sumEmbeddings(E_d), e_j) > \lambda$ **then**
        $set = set \cup e_j$
    **if** $set \neq \emptyset$ **then**
      $\Omega_i = set$
    **if** $|set| = 1$ **then**
      $\Gamma(i) = \Omega_i$; $E_d = E_d \cup \Omega_i$

`// High Probability Candidate Linking`

CreateDisambiguationGraphAndSolvePageRank$(\Omega_i, E_d)$; Rank candidates.

Select highest PR score $h$, second highest PR score $s$, average PR score $avg$.

**for** $m_i \in M$ *and* $|\Omega_i| > 1$ **do**
  **if** $s < dynThreshold$ **then**
    $\Gamma(i) = getEntityOf(h)$; $\Omega_i = getEntityOf(h)$; $E_d = E_d \cup \Omega_i$;
  **else**
    $\Omega_i = selectTop4RankedCandidates$

`// Final Linking and Abstaining`

CreateDisambiguationGraph$(\Omega_i, E_d)$

**for** $m_i \in M$ *and* $|\Omega_i| > 1$ **do**
  Perform PR and rank candidates, Select PR scores $h$, $s$ and $avg$.
  **if** $s < abstainingThreshold$ **then**
    $\Gamma(i) = getEntityOf(h)$; $\Omega_i = getEntityOf(h)$; $E_d = E_d \cup \Omega_i$;
  **else**
    $\Gamma(i) = NIL$; $\Omega_i = \emptyset$
  updateGraph$(\Omega_i, E_d)$

ranked candidate does not exceed the threshold *dynThreshold*, the highest ranked entity denotes the target entity of its surface form. In other words, if the relevance score margin between the highest ranked candidate and the other candidates is large, then the likelihood of the top-ranked candidate being the correct target entity is also high. If the threshold is exceeded, we reduce the candidate set of the respective surface form to the top-4 ranked candidate entities.

***Final Linking and Abstaining***

The last step *Final Linking and Abstaining* links the remaining entities or abstains if the algorithm is uncertain about the correct target entity (Lines 25-32). We first create an EL graph (cf. Section 8.2.5) and, then, iteratively link the entities of the remaining surface forms. For this purpose, every iteration applies the PageRank algorithm to the underlying graph and ranks the candidate entities of each surface form in descending order. The scores $h$, $s$, and $avg$ are calculated as in the previous step. The abstaining threshold *abstainingThreshold* is calculated using formula 8.1 with a different margin parameter ($margin_2$). If the second ranked candidate entity exceeds the abstaining threshold *abstainingThreshold*, the algorithm returns the *NIL* identifier for the respective surface form. Otherwise, the top ranked candidate entities denotes the target entity. After every iteration, we update the graph according to the changes in candidates and disambiguated entities and proceed until all surface form have been processed.

We note, that we apply the PageRank only once in step 3 due to performance reasons. The EL graph in step 4 usually does not include many candidate entities and, thus, we apply the PageRank in every iteration, also to provide the maximum accuracy in the abstaining task. The *margin* parameter to compute the *high probability threshold* and *abstaining threshold* varies in both steps. Information about the parameter choice is presented in Section 8.4.1.

### 8.2.4 Candidate Generation

In the first step, the goal is to reduce the number of possible candidate entities for each input surface form $m_i$ by determining a set of relevant target entities. We proceed as follows:

First, we search for all those entities that have already been annotated with $m_i$ in our previously constructed EL index. All entities that provide an exact surface form matching serve as candidate entities. If the candidate set is empty, we additionally use the candidate generation approach proposed by Usbeck et al. for AGDISTIS [Usb15]. This approach includes String normalization and String comparison via trigram similarity. The corresponding parameters are adopted from the default settings in the AGDISTIS framework.

Gathering all relevant candidate entities might result in a long list of candidates. To keep the list short and to improve the efficiency, we prune noisy candidates according to the following three criteria:

- **Prior probability:** In our work, the Sense Prior $p(e_j|m_i)$ estimates the probability of seeing an entity with a given surface form. If no Sense Prior is available due to non-available document-centric KBs, we use the Entity Prior $p(e_j)$ instead. We

select the top-$x$ entities as the candidates to keep the popular candidates. Both prior probabilities were precomputed and are stored in our index.

- **Context similarity:** We select the top-$x$ entities ranked by their context matching. To this end, we compute the cosine similarity between the entity-context embeddings and the Doc2Vec inferred context vector of the surface form.
- **Entity-topic similarity:** If a document contains at least two surface forms that have already been linked ($|E_d| > 1$), we create a topic vector $tv = \sum_{e_j \in E_d} vec(e_j)$. Variable $vec(e_j)$ denotes the entity embedding of $e_j$ and $E_d$ is the set of already linked entities. In the following, we select those remaining candidates of each surface form where the cosine similarity between the candidate entity embedding and $tv$ exceeds the *CandidateFilter* threshold.

For all criteria we use $x = 8$, which is enough to capture the relevant candidate entities. An experimental increase of $x$, resulted in a negligibly higher recall of the candidate generation task, but decreases EL accuracy and performance.

### 8.2.5 Entity Linking Graph and PageRank

In our approach, we generate an EL graph twice in order to link high probability candidate entities first and to perform abstaining afterwards. On this graph, we perform a random walk and determine the entity relevance, which can be seen as the average number of its visits. The random walk is simulated by a PageRank algorithm that permits edge weights and non-uniformly-distributed random jumps [Bri98; Whi03].

First, we create a **complete, directed** $K$-partite graph whose set of nodes $V$ is divided in $K$ disjoint subsets $V_0, V_1,...,V_K$. $K$ refers to the number of surface forms $S$ and $V_i$ is the set of generated candidate entities $\{e^i_1,...,e^i_{|V_i|}\}$ for surface form $m_i$. We define $m_0$ as pseudo surface form and use the subset $V_0 = \{e^0_1\}$ to contain the topic node. The topic node represents the average topic of all already linked entities in $E_d$. Hence, the edge weight between an entity $e^i_j$ and the topic node $e^0_1$ represents the relatedness between $e^i_j$ and all already linked entities. Since our graph is $K$-partite, there are only directed, weighted edges between candidate entities that belong to different surface forms. Connecting the entities that belong to the same surface form would be wrong since the correct target entities of surface forms are determined by the other surface forms' candidate entities (coherence).

The edge weights in our graph represent entity transition probabilities (ETP), which describe the likelihood to walk from a node to the adjacent node. We compute these probabilities by first computing the *Transition Harmonic Mean* (THM) between two nodes. The THM is the harmonic mean between two nodes' **entity relatedness** and the **context similarity** of the target entity (cf. Equation 8.2).

The entity relatedness between two nodes (entities) is the cosine similarity ($cos$) of the entities' semantic embeddings (vectors) $vec(e^i_j)$ and $vec(e^h_k)$. The semantic embedding of our topic node $e^0_1$ is the sum of all entity embeddings in $E_d$ (i.e., $vec(e^0_1) = \sum_{e_j \in E_d} vec(e_j)$). The context similarity between the target entity $e^h_k$ and the surrounding context of its surface form $m_h$ is the cosine similarity of $e^h_k$'s entity-context embedding $cvec(e^h_k)$, and the

inferred surrounding context vector $cvec(m_h)$ of $m_h$. In case, the target entity is our topic node the context similarity equals 0. The ETP is computed by normalizing the respective THM value (cf. Equation 8.3).

$$THM(e_j^i, e_k^h) = \frac{2 \cdot cos(vec(e_j^i), vec(e_k^h)) \cdot cos(cvec(e_k^h), cvec(m_h))}{cos(vec(e_j^i), vec(e_k^h)) + cos(cvec(e_k^h), cvec(m_h))} \tag{8.2}$$

$$ETP(e_j^i, e_k^h) = \frac{THM(e_j^i, e_k^h)}{\sum_{l \in (V \setminus V_i)} THM(e_j^i, l)} \tag{8.3}$$

Given the current graph, we additionally integrate a possibility to jump from any node to any other node in the graph during the random walk with probability $\alpha$. Typical values for $\alpha$ (according to the original paper [Whi03]) are in the range $[0.1, 0.2]$. We compute a probability for each candidate entity being the next jump target. Again, we either deploy the Sense Prior probability located in our EL index or the Entity Prior probability as jump probability for each node (entity). The Entity Prior probability is used if no document-centric KBs are available. The probability to jump to or from the topic node equals 0.

Figure 8.2 shows a possible candidate entity graph. The surface form 'TS' has only one candidate entity and consequently has already been linked to the entity *Time Square*. The surface form 'New York' is still ambiguous, providing two candidates. The topic node $e_1^0$ comprises the already disambiguated surface form 'Time Square'. We omit the edge weights and jump probabilities in the figure to improve visualization.

After constructing the EL graph, we apply the PageRank algorithm and compute a

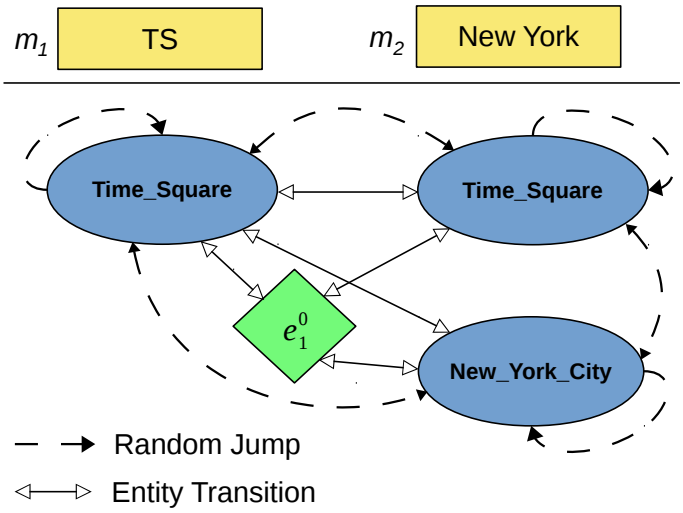



**Figure 8.2:** Example EL graph with candidates for the surface forms 'TS' and 'New York' and a topic vector. Solid lines denote entity transition probabilities and dashed lines denote jump probabilities between entity pairs.

relevance score for each candidate entity. Depending on the EL task, our approach decides which candidate entity is the correct target entity or abstains if no appropriate candidate is available (cf. Algorithm 8.2.3).

## 8.3 Data Sets

To evaluate DoSeR on general-domain entities, we make use of the same data sets as proposed in Section 6.4 on Page 109. All these data sets are integrated in the online EL evaluation framework GERBIL by default. Further, when we evaluate DoSeR in the biomedical domain, we use the CalbCSmall and CalbCBig corpus as training corpus and test data set, similar to Chapter 5. For an in-depth data set description, we refer to Section 5.4.

Apart from natural language text data sets, we also investigate how DoSeR links entities in tables. For this purpose, we use six data sets from different domains whose entities are contained in Wikipedia. An overview of the data set statistics is given in Table 8.1.

1. **Wiki-Manual:** Limaye et al. [Lim10] created a small data set of 36 Wikipedia tables extracted from Wikipedia article texts (non-Infobox tables). Some columns overlap with the Web-Manual data set.
2. **Web-Manual:** A set of 371 web tables was crawled by Limaye et al. [Lim10]. The difference between Wiki-Manual and Web-Manual is that the cell and header texts in the latter are noisier. The data set comprises a huge number of 51 898 cells, but only 9239 of them are annotated with ground truth entities.
3. **Wiki-Links:** This data set was specifically created to evaluate cell EL algorithms at large scale. The table set consists of Wikipedia tables where at least 90% of the cells internally link to entities in Wikipedia [Lim10].
4. **LimayeAll:** The LimayeAll data set was re-created in the context of the table annotation approach *TableMiner* by Zhang et al. [Zha14]. The authors re-created the Limaye et al. [Lim10] data sets Wiki-Manual, Web-Manual and Wiki-Links to correct wrong or changed Wikipedia annotations, and combined them. In addition, it was assumed that the original ground truth annotations of the data sets are very sparse and possibly biased. Thus, the authors changed a huge number of surface forms to complicate the EL process.
5. **IMDb:** The IMDb data set, also created in the context of the table EL approach *TableMiner* [Zha14], contains 7416 tables randomly extracted of the IMDb movie website. Each movie web page contains a table listing the actors/actresses and the corresponding characters played.
6. **MusicBrainz:** Our last data set MusicBrainz comprises about 1400 tables which were randomly extracted from the MusicBrainz record label web pages by Zhang et al. [Zha14]. Typically, a web page lists the music released by a production company. A table has about 8 columns with one listing the music release titles and one listing the respective artists.

We note that the table data sets listed above are exclusively annotated with named entities (i.e., persons, locations and organizations). Basically, we could have omitted all candidate entities that do not belong to these types during our experiments to further improve the underlying results. However, since we do not adapt our approach or EL index to specific data sets, we have used the same general-domain entity index (Wikipedia) for all data sets.

**Table 8.1:** Table data set statistics

| Data Set | #Tables | #Average Rows | #Average Columns | #Entity Annotations |
|---|---|---|---|---|
| Wiki-Manual | 36 | 37 | 4 | 1691 |
| Web-Manual | 371 | 35 | 2 | 9239 |
| Wiki-Links | 6085 | 20 | 3 | 131 807 |
| LimayeAll | 6310 | 22 | 110 | 231 657 |
| IMDB | 7416 | 14 | 1 | 66 564 |
| MusicBrainz | 1406 | 78 | 2 | 93 110 |

## 8.4 Evaluation

In our evaluation, we show that DoSeR achieves state-of-the-art results across different domains and document structures and types. Before we report the results in detail, we first describe the experimental setup in Section 8.4.1. Next, we present how DoSeR performs on linking entities from general-domain KBs in news documents, RSS-feeds, tweets and tables in Section 8.4.2 and 8.4.3. It follows the evaluation on how DoSeR performs in the biomedical domain in Section 8.4.4. In Section 8.4.5, we analyze the EL results after enabling the abstaining mechanism in our algorithm. Finally, we present a parameter study of our semantic embeddings in terms of Word2Vec and Doc2Vec architectures and their optimal dimensions in Section 8.4.6.

### 8.4.1 Experimental Setup

The DoSeR framework is fully-implemented in Java and Python. For the Word2Vec and Doc2Vec algorithms, we chose Gensim [Řeh10], a robust and efficient framework to realize unsupervised semantic modeling from plain text. Before our algorithm is able to link entities, we first have to perform some preprocessing steps. First, we choose a set of KBs whose entities define our target entity set $\Omega$. When we disambiguate general-domain entities (as in Section 8.4.2 and 8.4.3), we make use of the current version of DBpedia (v.2015-10) as entity database (i.e., core KB). This version reflects information from the last years Wikipedia version. Overall, we extracted $\approx$ 4.1 million entities (all entities belonging to the *owl:thing* class) out of DBpedia that we would like to link in our work. Next, we selected Wikipedia ($\approx$ 81 million annotations) and the Google Wikilinks Corpus ($\approx$ 40 million annotations) as entity-annotated document KBs that serve as training data for our semantic entity embeddings (Word2Vec). To create the Doc2Vec entity-context embeddings, our framework parses the entities' Wikipedia pages and removes all Wikipedia syntax elements as well as tables. The resulting natural language text documents serve as

input for the Doc2Vec algorithm. We note that in contrast to Chapter 7, DoSeR does not subdivide the entity texts into paragraphs to increase the performance of our approach.

In Section 8.4.4, we evaluate DoSeR on the biomedical data sets CalbcSmall and CalbcBig. To create our entity database, we again (similar to Chapter 5) focus on the four major namespaces UMLS, Disease, Uniprot and EntrezGene in both CalbC data sets. Here, we use the original entity-annotated CalbC documents and crawled the respective entity-centric KBs in the LOD cloud (i.e., LinkedLifeData, Uniprot, NCBI) to gather the respective entity information. More information about the CalbC data sets can be found in Chapter 5.

In the following, DoSeR learns entity embeddings and entity-context embeddings with Word2Vec and Doc2Vec. To train the entity embeddings with Word2Vec, we defined a feature space of $d = 400$ dimensions. DoSeR typically employs the skip-gram architecture that performs better with infrequent words [Mik13a]. In terms of Doc2Vec, we defined a feature space of $d = 1000$ dimensions. DoSeR learns the entity-document embeddings with the PV-DM architecture. An experimental comparison between the architectures and various settings for parameter $d$ is presented in Section 8.4.6. The Word2Vec training time took $\approx 90$ minutes on our personal computer with a 4x3.4GHz Intel Core i7 processor and 16 GB RAM (1 corpus iteration). The training time for Doc2Vec took $\approx 2$ days on our server with 20 cores and 25 GB RAM with 5 iterations overall.

Our approach offers several parameters to tweak the results. In the following, we will mention only those that have the most impact on the results.

- **Surrounding Context:** For Doc2Vec, DoSeR uses a surrounding context of 200 words, which denotes that 100 words before and after the surface forms form the context. Using more context words, results in less meaningful query vectors (cf. Chapter 7).

- **Candidate Filter:** The cosine similarity ranges from -1 (unequal) to 1 (equal). A reasonable way to tune $\lambda$ is to sweep the value between $0.25 < \lambda < 0.8$ (necessary similarity). We selected the value $\lambda = 0.57$ according to the best averaged F1 values throughout the experiments.

- **PageRank:** DoSeR performs 100 PageRank iterations since the overall results do not change with more iterations. In terms of the PageRank jump probability $\alpha$, we chose $\alpha = 0.1$ in algorithm step 3 (according to the original paper [Whi03]). In algorithm step 4, we chose $\alpha = 0.2$ to increase the prior influence (i.e., a robust baseline) since the correct entity could not be determined with the help of topical coherence in the steps before. In the disambiguation step *High Probability Candidate Linking*, we determined the parameter $margin_1 = 0.5$ by sweeping the value between $0.2 < margin_1 < 0.6$. Again, the best value was selected according to the best averaged F1 values throughout the experiments.

- **Abstaining:** We note that abstaining is disabled by default using $margin_2 = -\infty$. To provide the best abstaining results, we chose $margin_2 = 0.3$ by sweeping the value between $0.2 < margin_2 < 0.6$ as described above.

### 8.4.2 Entity Linking Results on General-Domain Knowledge Bases

In the following, we directly compare our approach to publicly available, state-of-the-art EL systems, which disambiguate Wikipedia, DBpedia or YAGO entities, via GERBIL v1.1.4 (D2KB task). Our comparative systems are the currently available versions of, AIDA [Hof11], Babelfy [Mor14], WAT [Pic14] and Wikifier [Che13; Rat11]. Wikifier and WAT use Wikipedia as underlying KB and link surface forms directly to Wikipedia pages. Babelfy also returns Wikipedia entities but uses BabelNet as KB, which was automatically created by linking Wikipedia to WordNet [Fel98]. In contrast, AIDA relies on the entity-centric KB YAGO2, while additionally making use of Wikipedia knowledge. For all systems we chose the best configurations according to the authors. Moreover, we downloaded the Wikifier and AIDA systems (new index) and installed both systems on our server using the 'Full Gurobi Configuration' for Wikifier and 'CocktailParty Configuration' for AIDA (WAT and Babelfy are integrated in GERBIL).

In addition to these frameworks, we define the strong baseline *Sense Prior* that links surface forms to the entities with the highest prior probability (cf. Section 3.2.2). We also present the results when excluding the entity-context embeddings (denoted as *DoSeR (W2V)*). We investigate how well the approach performs with entity-embeddings as entity relatedness feature only. In this case, we use the entity embeddings directly to compute the ETP (cf. Section 8.2.5).

Table 8.2 shows the micro-averaged precision, recall and F1 values in comparison to the competitor systems on all data sets. The corresponding GERBIL result sheet is available on the GERBIL website[1] and can be used to make comparisons to our approach in future evaluations.

Overall, our approach attains the best averaged F1 value of all systems. Thereby, it outperforms Wikifier by 5 F1 percentage points on average. Additionally, we significantly outperform the other systems as well as the Sense Prior baseline by up to 25 F1 percentage points on average. On the data sets ACE2004, MSNBC, Microposts2014-Test and N3-Reuters our approach performs exceptionally well (up to 12 F1 percentage points in advance). We note that our Sense Prior baseline outperforms Wikifier on the Microposts2014-Test data set because of using a newer version of Wikipedia. The Micropost2014-Test data set was released in 2014 and obviously queries some very new (or changed) entities. On the DBpedia Spotlight and N3-RSS-500 data sets our approach also performs best with F1 values of $\approx 0.81$ (DBpedia Spotlight) and $\approx 0.75$ (N3-RSS-500) respectively. Considering the AIDA/CONLL-TestB data set, our approach performs slightly better than Wikifier but performs comparatively poor with a F1 value of $\approx 0.78$ compared to $\approx 0.84$ by the WAT system. The reasons for this are two-fold: First, the underlying data set is still annotated with entities whose identifiers have been changed over the years with updates. Thus our service returns wrong entity URLs according to the ground truth. The same problem occurs in the AIDA system when using the newer AIDA entity index. In this case, the F1 value drops from 0.82 to 0.77. In an experiment where we disambiguate the original AIDA entities, our system achieves a F1 value of $\approx 0.84$. Second, a more detailed analysis

1 http://dx.doi.org/10.5281/zenodo.51250, last accessed on 2016-11-28

**Table 8.2:** Micro-averaged precision, recall and F1 values of DoSeR, DoSeR without Doc2Vec, the prior probability baseline, Wikifier, AIDA, Babelfy and WAT on nine data sets

**Precision**

| Data Set | DoSeR | DoSeR (W2V) | Sense Prior | Wikifier | AIDA | WAT | Babelfy |
|---|---|---|---|---|---|---|---|
| ACE2004 | **0.912** | 0.880 | 0.838 | 0.824 | 0.850 | 0.846 | 0.694 |
| AIDA-TestB | 0.784 | 0.754 | 0.662 | 0.777 | 0.775 | **0.852** | 0.809 |
| AQUAINT | 0.847 | 0.847 | 0.805 | **0.862** | 0.571 | 0.808 | 0.773 |
| DBpedia Spot. | **0.814** | 0.780 | 0.749 | 0.797 | - | 0.686 | 0.583 |
| IITB | 0.744 | 0.741 | 0.714 | **0.767** | 0.287 | 0.647 | 0.653 |
| Micro.2014 | **0.783** | 0.737 | 0.660 | 0.576 | 0.514 | 0.662 | 0.640 |
| MSNBC | **0.913** | 0.881 | 0.714 | 0.892 | 0.800 | 0.824 | 0.804 |
| N3-Reuters128 | **0.856** | 0.817 | 0.705 | 0.703 | 0.679 | 0.734 | 0.685 |
| N3 RSS-500 | 0.752 | 0.715 | 0.679 | 0.732 | 0.743 | 0.711 | **0.770** |

**Recall**

| Data Set | DoSeR | DoSeR (W2V) | Sense Prior | Wikifier | AIDA | WAT | Babelfy |
|---|---|---|---|---|---|---|---|
| ACE2004 | **0.901** | 0.864 | 0.824 | 0.824 | 0.783 | 0.759 | 0.611 |
| AIDA-TestB | 0.784 | 0.754 | 0.661 | 0.777 | 0.774 | **0.836** | 0.794 |
| AQUAINT | 0.838 | 0.838 | 0.801 | **0.862** | 0.499 | 0.732 | 0.682 |
| DBpedia Spot. | **0.806** | 0.770 | 0.742 | 0.797 | - | 0.621 | 0.470 |
| IITB | 0.738 | 0.735 | 0.708 | **0.763** | 0.256 | 0.579 | 0.514 |
| Micro.2014 | **0.719** | 0.674 | 0.604 | 0.576 | 0.405 | 0.542 | 0.385 |
| MSNBC | **0.908** | 0.871 | 0.708 | 0.814 | 0.765 | 0.735 | 0.756 |
| N3-Reuters128 | **0.844** | 0.803 | 0.695 | 0.704 | 0.531 | 0.573 | 0.502 |
| N3 RSS-500 | **0.750** | 0.711 | 0.677 | 0.732 | 0.689 | 0.655 | 0.653 |

**F1**

| Data Set | DoSeR | DoSeR (W2V) | Sense Prior | Wikifier | AIDA | WAT | Babelfy |
|---|---|---|---|---|---|---|---|
| ACE2004 | **0.907** | 0.872 | 0.831 | 0.824 | 0.815 | 0.800 | 0.650 |
| AIDA-TestB | 0.784 | 0.754 | 0.661 | 0.777 | 0.774 | **0.843** | 0.802 |
| AQUAINT | 0.842 | 0.842 | 0.803 | **0.862** | 0.533 | 0.768 | 0.725 |
| DBpedia Spot. | **0.810** | 0.775 | 0.745 | 0.797 | - | 0.652 | 0.520 |
| IITB | 0.741 | 0.738 | 0.711 | **0.765** | 0.270 | 0.611 | 0.576 |
| Micro.2014 | **0.750** | 0.704 | 0.630 | 0.576 | 0.453 | 0.595 | 0.480 |
| MSNBC | **0.911** | 0.876 | 0.711 | 0.851 | 0.782 | 0.777 | 0.779 |
| N3-Reuters | **0.850** | 0.810 | 0.700 | 0.694 | 0.596 | 0.644 | 0.579 |
| N3 RSS-500 | **0.751** | 0.713 | 0.678 | 0.732 | 0.716 | 0.682 | 0.707 |
| **Average** | **0.816** | 0.787 | 0.718 | 0.764 | 0.617 | 0.708 | 0.646 |

of the surface forms' textual context is necessary to perform even better. Nevertheless, our algorithm outperforms the other systems and also AIDA which was optimized on this data set. Regarding AQUAINT and IITB, Wikifier leads DoSeR by 2 percentage points F1 on both data sets.

In order to evaluate how DoSeR performs with significantly less entity data, we assume Wikipedia to be our single KB and to have significantly less annotations. To this end, we computed the Sense Prior and our entity embeddings with omitting 80% Wikipedia entity annotations during training. The omitted annotations were selected randomly. Further, we assume to have much less entity describing information. As a consequence, we exclusively trained our entity-context embeddings on the introducing sentence of a Wikipedia entity. Given this setting, DoSeR still achieves $\approx 73$ F1 percentage points on average across all data sets. Further reducing the amount of entity annotations (i.e., omitting 90% Wikipedia annotations) led to an average F1 value of $\approx 0.70$ across all data sets. This shows, that our approach provides consistent results despite significantly less entity describing data in form of entity annotations and entity descriptions.

In **summary**, we state that our approach significantly outperforms other publicly available EL approaches. Overall, our approach disambiguates the entities highly accurate and attains state-of-the-art or nearly state-of-the-art results on all nine data sets. Hence, our approach is very well suited for all kinds of documents available in the web (e.g., tweets, news, etc.). In terms of performance, our approach annotates roughly as fast as the Wikifier and WAT annotation system but is slower compared to Spotlight and AIDA. The Babelfy system is the slowest and takes too much time, especially on the IITB data set. Our system has the advantage to accept multiple queries in parallel, but is not yet optimized for high-performance EL.

***Further Discussions***

Comparing our results to those of other state-of-the-art approaches that are not publicly available is not an easy task. Reimplementing the respective algorithms is not an absolutely fair method to compare the approaches with our KB: Usually crucial implementation details remain unknown in the original publications, since the focus mostly lies on the algorithm instead of the implementation.

Anyhow, we use the work of Guo et al. [Guo14] as an entry point in the following. Their approach was exclusively evaluated and optimized on the ACE2004, MSNBC and AQUAINT data sets on which the authors achieved state-of-the-art results. A direct comparison of our results and the results of [Guo14] shows that both works perform equally well on the MSNBC data set. Furthermore, our approach performs better on the ACE2004 data set (0.906 vs. 0.877 F1) but loses on the AQUAINT data set (0.842 vs. 0.907 F1). The problem with a pure number-based comparison, however, lies in the uncertainty in the underlying KB used in the experiments. If the underlying KB has a lower number of entities, the average likelihood of a wrong entity assignment is also reduced. In order to compare our algorithm to the approach by Guo et al. [Guo14], we introduce the concept of the Surface Form Ambiguity Degree (SFAD). The concept is based on the following two assumptions:

- Both approaches are able to disambiguate all entities in the ground truth data set, i.e., the KB covers the entities in the data sets and contains similar entity occurrences resulting from a given corpus (important for prior computation).
- The candidate entities retrieved from the KB contain the correct entity, i.e., the error introduced by candidate selection is zero.

Under these assumptions, a varying prior probability of an entity defines the degree of ambiguity, the SFAD, for that surface form. So the SFAD describes how many entities are potentially relevant for a specific surface form. Since our approach has a (significantly) lower prior probability on these data sets, the SFAD is higher, respectively. In Table 8.3, we compare the differences of the best result and the result achieved with the Prior alone for our approach and the Guo et al. approach.

**Table 8.3:** Differences of the best result and the prior when using DoSeR and Guo et al. [Guo14]

| **Approach** | **ACE2004** | **MSNBC** | **AQUAINT** |
|---|---|---|---|
| DoSeR F1(best) - F1(prior) | 0.076 | 0.200 | 0.039 |
| Guo et al. F1(best) - F1(prior) | 0.022 | 0.049 | 0.035 |
| $\Delta$ in F1 percentage points | **5.4** | **15.1** | **0.4** |

Overall, our EL index contains more entities that are relevant for a surface form on average and hints that our core-algorithm (without KB and candidate selection) is more robust than the approach from Guo et al. [Guo14]. Another evidence is that the authors re-implemented the approach used in Wikifier and achieved significantly better results on their KB as we achieve with GERBIL with Wikifier's original KB. Guo et al. also reported the results of former, well-known state-of-the-art approaches (e.g., Cucerzan [Cuc07], Han et al. [Han11b], Glow [Rat11]), but we do not discuss the results in detail because these approaches perform worse than Wikifier and the approach of Guo et al.

Considering the IITB data set, the system by Han et al. [Han12] performs best with a micro F1 value of 0.80. The authors did not evaluate their system on other data sets. However, their topic model approach is fully-trained on Wikipedia and takes all words into account. Since, the IITB data set consists of long documents very similar to those in Wikipedia, the system performs best on it.

In 2014, the Micropost2014-Test data set was created in the context of the workshop challenge *Making Sense of Microposts* [Bas14]. The best system in the workshop was proposed by Microsoft, which attains a micro F1 value of 0.70. To the best of our knowledge, this has been the best EL approach on this data set so far, but is outperformed by our approach by $\approx 5$ percentage points.

Considering the AIDA CONLL-TestB data set, the current state-of-the-art approach has been presented by Huang et al. [Hua15] and attains a micro F1 value of 0.866. Similar to our approach, the authors learned semantic embeddings with a deep neural network approach from DBpedia and Wikipedia (but not with Word2Vec and Doc2Vec). Again, the approach was only evaluated on the AIDA CONLL-TestB data set as well as on a

tweet data set. Experiments show that we can also further improve our results on this data set to a micro F1 value of 0.850 when we perform the following changes:

- Reducing candidate entities (lower SFAD)
- Training the semantic embeddings on DBpedia instead of Wikipedia
- Using an older entity index

However, since our main goal was to create a **robust** EL approach that performs well on several data sets with varying underlying document properties, we did not optimize the DoSeR algorithm on a single data set.

A very recent proposed state-of-the-art approach is the *Probabilistic Bag-of-Hyperlinks-Model* (PBoH) by Ganea et al. [Gan16] (Section 3.2.4). Similar to the other mentioned works, the implementation of the PBoH-EL system is not publicly available. However, the authors provided the respective GERBIL v.1.2.2 result sheet of their evaluation[1]. Since we use GERBIL v.1.1.4 in all experiments by default, the results are not directly comparable because the authors of GERBIL performed minor data set and system changes in each update. More information about different GERBIL versions are available on the respective web page. However, to compare DoSeR to the PBoH model we also evaluated DoSeR on the current version 1.2.4 and report the 'GSinKB Micro F1 scores', i.e., queries with NIL ground truth annotations are omitted. Unfortunately, version 1.2.2 is not available any more. Version 1.2.4 is an experimental version that consumes very much time for evaluation and showed some bugs with DoSeR during the experimental runs.

Figure 8.3 contrasts the F1 values of DoSeR evaluated with GERBIL v.1.1.4 and v.1.2.4 and PBoH evaluated with GERBIL v.1.2.2. DoSeR (significantly) outperforms PBoH on eight out of nine data sets. Regarding the AIDA CONLL-TestB data set the story looks different. PBoH significantly outperforms DoSeR by $\approx 8$ percentage points F1.

### 8.4.3 Entity Linking Results on Tables

In this section, we evaluate DoSeR on table data sets and compare the results to state-of-the-art table EL systems that link Wikipedia entities. These include the collective Semantic Message Passing framework by Mulwad et al. [Mul13] (*Mul-Col*), the collective approach by Limaye et al. [Lim10] (*Lim-Col*), TableMiner+ by Zhang [Zha16] (*TM+*) as well as the baseline algorithms Least Common Ancestor (*Lim-LCA*) and Majority Vote (*Lim-Maj*) [Lim10; Zwi13a]. LCA links entities according to the least common ancestor type. Given a table column, LCA links those candidate entities of the column cells that belong to the same entity type. The majority baseline algorithm first selects the entity type that occurs most often across all candidate entities of all cells within a column. Then, it links all candidate entities that belong to this specific type. For an algorithmic description of the baseline algorithms, we refer to the original work [Lim10]. For the other approaches, we refer to our related work Chapter 3 and the respective references. We emphasize that, apart from the baseline algorithms, all those systems link entities to table cells with the help of a collective approach where cells, columns (with types) and relations between columns

1 http://gerbil.aksw.org/gerbil/experiment?id=201604270015, last accessed on 2016-11-28

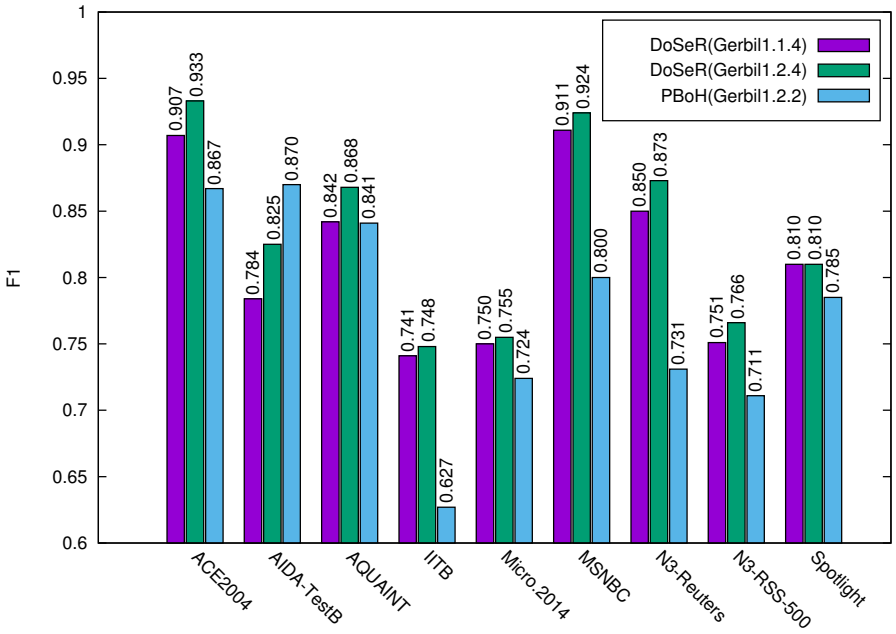



**Figure 8.3:** Micro-averaged F1 values of DoSeR and the Probabilistic Bag-of-Hyperlinks Model with different GERBIL versions

are annotated simultaneously. Unfortunately, these systems are not publicly available. Thus, we use the values described in the original paper of the respective data sets. In contrast to the previous section, we did not use GERBIL to determine the result values since the underlying data sets are not integrated. Instead, we manually downloaded the data sets and used the given surface forms as input for our system. We concatenated the content of additional cells within the same row to create the surrounding context.

Table 8.4 shows the micro-averaged F1, precision and recall values of DoSeR on our table data sets. Further, Table 8.5 shows the micro-averaged F1 values of DoSeR and other table EL systems. Regarding the old Limaye et al. [Lim10] data sets, DoSeR achieves the best results. We assume that our results would have been even better if we had sorted out all non-existing ground truth annotations (e.g., entities whose identifiers have changed over the years). In fact, on the Wiki-Links data set (comprising original Wikipedia data) we can see that our embeddings reliably capture the knowledge located within Wikipedia (F1 value of $\approx 0.963$). In addition, these experiments emphasize that collectively annotating table cells, columns and relations (as done by all competitive systems) does not necessarily lead to the best results. The TableMiner+ approach has not been evaluated on these old data sets. Instead, Zhang [Zha16] evaluated TableMiner+ on the revised LimayeAll data set, where TableMiner+ tops DoSeR by $\approx 0.7$ F1 percentage points. On the IMDB data set, DoSeR and TableMiner+ perform exceptionally well with a F1 value of $\approx 0.98$. In contrast, on the MusicBrainz data set, DoSeR significantly outperforms TableMiner+ by $\approx 10$ F1 percentage points.

In **summary**, we showed that DoSeR links surface forms in tables highly accurate and outperforms other table annotation approaches on most data sets.

**Table 8.4:** Micro-averaged F1, precision and recall values of DoSeR on 6 table data sets

| Data set | F1 | Precision | Recall |
|---|---|---|---|
| Wiki-Man | 0.869 | 0.880 | 0.857 |
| Web-Man | 0.861 | 0.874 | 0.849 |
| Wiki-Links | 0.963 | 0.968 | 0.958 |
| LimayeAll | 0.830 | 0.851 | 0.809 |
| IMDB | 0.987 | 0.990 | 0.985 |
| MusicBrainz | 0.953 | 0.957 | 0.948 |

**Table 8.5:** Micro-averaged F1 values of DoSeR, TableMiner+, Limaye-Collective, Limaye-Majority, Limaye-LCA and Mulwad-Collective on six data sets

| Data set | DoSeR | TM+ | Lim-Col | Lim-Maj | Lim-LCA | Mul-Col |
|---|---|---|---|---|---|---|
| Wiki-Man | **0.869** | - | 0.839 | 0.742 | 0.598 | 0.674 |
| Web-Man | **0.861** | - | 0.814 | 0.759 | 0.597 | 0.631 |
| Wiki-Links | **0.963** | - | 0.843 | 0.776 | 0.679 | 0.759 |
| LimayeAll | 0.830 | **0.837** | - | - | - | - |
| IMDB | **0.987** | 0.976 | - | - | - | - |
| MusicBrainz | **0.953** | 0.849 | - | - | - | - |

### 8.4.4 Entity Linking Results in the Biomedical Domain

In the previous section, we analyzed how DoSeR performs on general knowledge from Wikipedia and DBpedia. In this section, we evaluate our system on a specialized domain, namely the biomedical domain. Similar to Chapter 5, we use the CalbC for training and evaluation purposes. An in-depth description of the respective CalbC subcorpora CalbCSmall and CalbCBig can be found in Section 5.4. To provide a better comparison, we contrast the DoSeR results with those achieved with the federated Learning To Rank (LTR) approach proposed in Chapter 5. Since the LTR approach does not collectively link all surface forms within a document, we report the DoSeR results after collective and non-collective EL. In the collective configuration, our algorithm is not able to retrieve a ranked list of (correct) entity assignments for each surface form. As a consequence, to return multiple correct EL results, we modified our approach and returned the list of remaining candidate entities in the *Final Linking and Abstaining* step (cf. Section 8.2.3) sorted according to their PageRank score. In the non-collective configuration, the DoSeR algorithm relies on the Sense Prior probability and the textual context matching score (computed with Doc2Vec) and allows us to return a relevance-sorted entity list. In both approaches we return a list containing at most 10 entities (equally to the LTR approach in Section 5.5.3).

In our general-domain evaluation, we leveraged the Wikipedia article pages as Doc2Vec training corpus. In CalbC, the documents do not describe entities as it is the case in Wikipedia. For that reason, we created our entity-context embeddings based on the

surrounding context of annotated entities (cf. Section 7.2). Hereby, we used a context window of 100 words before and after the surface form during the training phase (as suggested in Chapter 7).

Since the CalbC provides multiple correct entity annotations per surface form, we report the mean reciprocal rank (MRR), recall and mean average precision (MAP) in this evaluation. All these measures were averaged over 5-fold cross validation runs. For every cross-validation run, we used the unified set of the 4 training partitions to train our entity embeddings (i.e., entity embeddings and entity-context embeddings).

Figure 8.4 shows the MRR, recall and MAP values of DoSeR (collective and non-collective) and the federated LTR approach on the CalbCSmall data set. Overall, the non-collective approach of DoSeR performs worse than our LTR approach. Obviously, our LTR feature set is superior ($\approx 4-6$ percentage points on our measures) to the DoSeR feature set only comprising the Sense Prior and surrounding context matching with Doc2Vec. By contrast, our collective approach achieves the best results overall with outperforming the LTR approach. A MRR of 0.937 indicates a high level of reliability in terms of ranking a correct entity on top. In terms of recall and MAP, DoSeR-collective tops the LTR approach by $\approx 3$ percentage points. An evaluation on the CalbCBig data set results in nearly the same result values for all approaches.

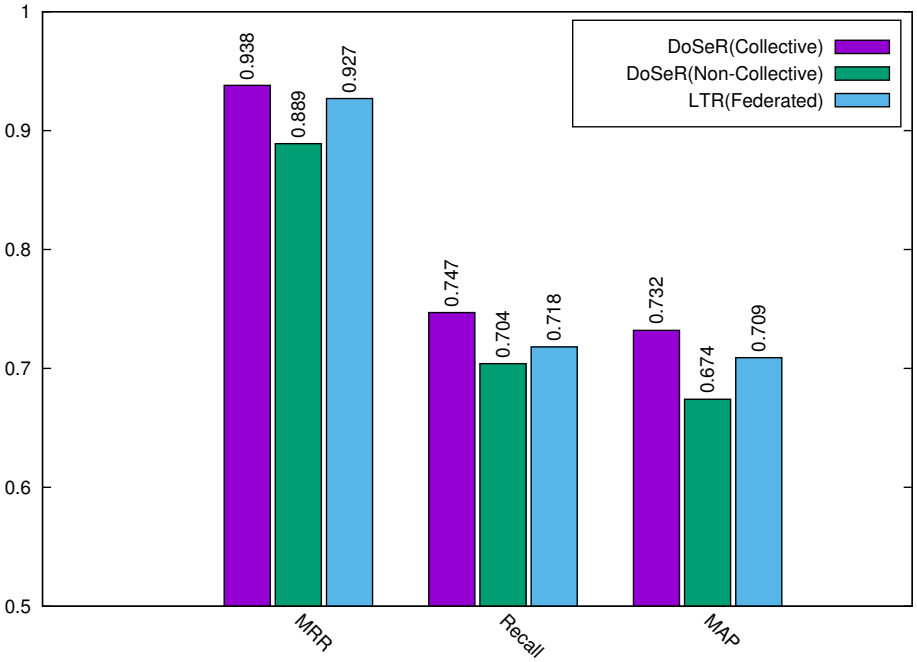



**Figure 8.4:** MRR, recall and MAP values of DoSeR (collective), DoSeR (non-collective) and the federated LTR approach on CalbCSmall

We also conducted an experiment with our default DoSeR settings as used in the general-domain experiments. Here, we analyzed whether the retrieved entity (only one entity is retrieved by default) is located among the ground truth entity list. Using the 0-1 loss, i.e., we lose a point if we get a wrong entity, we obtain an accuracy value of $\approx 0.871$. When we

apply the same measure on our LTR approach, we obtain an accuracy value of $\approx 0.848$.

We **summarize**, that DoSeR outperforms our federated LTR approach and performs well in the biomedical domain. Although, the LOD cloud lacks relevant entity data for EL [Zwi13b], DoSeR is able to leverage the evidences in form of annotated entities in the document-centric KB to provide strong EL results.

### 8.4.5 Abstaining

Abstaining is an important task in EL algorithms when it comes to link surface forms whose referent entity is not in the entity target set $\Omega$. It is also used if there is uncertainty about the correct entity due to insufficient context information.

In this experiment, our algorithm returns the pseudo-entity *NIL* in the following situations:

- If no candidate entities can be found during the candidate generation step (cf. Section 8.2.4).
- If the algorithm is uncertain about the correct entity after the last PageRank iteration (cf. Algorithm 8.2.3).

For experimental purpose, we downloaded the original IITB data set, which additionally contains 7652 *NIL* annotations in addition to the default annotations (18 897 annotations overall), and report the EL *accuracy*. We also rerun the GERBIL experiments with abstaining to investigate to what extent the results decrease on data sets which do not provide *NIL* ground truth annotations.

Conducting the experiment on the manually downloaded IITB data set resulted in an EL accuracy of 0.757 (micro-averaged). With returning 6120 *NIL* annotations overall, our algorithm does not find candidates for surface forms in 3823 cases ($\approx 62.5\%$) and abstains 2297 surface forms ($\approx 37.5\%$). When we tune our abstaining parameter to abstain more aggressive, our overall accuracy slightly decreases. Unfortunately, the authors of the topic-model, state-of-the-art approach [Han12] on this data set did not provide abstaining results for comparison in their work. However, Table 8.6 reports the micro F1 values of our algorithm with abstaining on all data sets in the GERBIL evaluation.

**Table 8.6:** F1 values of our approach with abstaining on data sets without *NIL* annotations

| **Data Set** | **F1** | **Change in F1 percentage points** |
|---|---|---|
| ACE2004 | 0.892 | -1.65 |
| AIDA-TestB | 0.782 | -0.26 |
| AQUAINT | 0.820 | -2.61 |
| DBpedia Spotlight | 0.773 | -4.57 |
| MSNBC | 0.906 | -0.55 |
| N3-Reuters128 | 0.809 | -4.82 |
| IITB | 0.722 | -2.56 |
| Microposts-2014 Test | 0.607 | -7.07 |
| N3 RSS-500 | 0.738 | -1.73 |

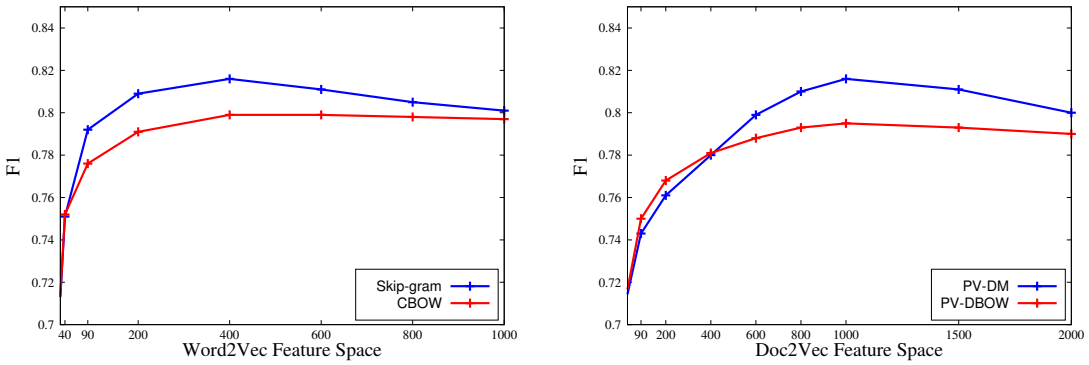
As a result of GERBIL not querying surface forms with *NIL* annotations in the ground truth, our results (slightly) decrease. Nevertheless, the number of abstained surface forms is very limited and, thus, our approach still outperforms Wikifier on 6 out of 9 data sets. On the Microposts2014-Test data set, the F1 decrease is the highest with 7 percentage points. Obviously, our algorithm is sometimes uncertain about the correct entity and abstains, which is due to a small number of surface forms per document. In other words, our algorithm lacks sufficient evidences about the correct entity and, hence, abstains due to exceeding the abstaining threshold.

In **summary**, we state that our algorithm is able to successfully abstain entity annotations if evidences about the correct entities are missing. Our abstaining mechanism performs well even if data sets do not provide *NIL* annotations (as simulated by GEBRIL).

### 8.4.6 Semantic Embeddings Parameter Study

The accuracy of our approach depends on a number of parameters, foremost the parameters of the semantic embeddings. In order to analyze this sensitivity, we conducted experiments in which we varied the number of dimensions of our semantic embeddings. We report the results for both Word2Vec and Doc2Vec architectures (i.e., CBOW vs. skip-gram and PV-DM vs. PV-DBOW). In this experiment, we used the same KBs and test data sets (GERBIL) as in Section 8.4.2.

Figure 8.5(a) depicts the micro-averaged F1 values (across all data sets) of our approach when using either the CBOW or skip-gram Word2Vec architecture and a specific number of dimensions. During this experiment, the corresponding Doc2Vec architecture was set to PV-DM since it is better suited as we will see in the following. Basically, in our experiments, the skip-gram architecture consistently created better entity embeddings than CBOW. This might be due to skip-gram performs better with infrequent words (entities) in the training corpus [Mik13a]. However, the difference between both architectures is $\approx 1-2$ percentage points F1. On $d = 400$ the result margin between both architectures is maximized and the average F1 value reaches its peak. It is interesting to see that even more dimensions slightly decrease the result values. We assume that this leads to some kind of overfitting and, thus, the optimal number of dimensions for entity embeddings probably depends on the number of entities and amount of training data.




**(a)** Word2Vec **(b)** Doc2Vec

**Figure 8.5:** Comparison of Word2Vec and Doc2Vec architectures with various dimensions

We conducted the same experiment for our entity-document embeddings (Doc2Vec). In this particular case, we used the skip-gram architecture as baseline training algorithm for the entity embeddings (Word2Vec). Figure 8.5(b) depicts the corresponding micro-averaged F1 values for various dimensions and both Doc2Vec architectures. The PV-DM architecture for Doc2Vec performs better if the number of dimensions is higher than $d = 400$. We assume that the context consideration in the PV-DM architecture leads to an advance. However, the best average F1 value is achieved with $d = 1000$, whereby the difference between PV-DBOW and PV-DM is at most $\approx 2$ percentage points F1.

In **summary**, we state that the skip-gram architecture for Word2Vec and the PV-DM architecture for Doc2Vec perform best in DoSeR. It is interesting to see that the number of optimal dimensions for entity embeddings must be geared to the underlying corpora.

## 8.5 Conclusion

In this chapter, we presented DoSeR, a robust EL framework that is KB-agnostic in terms of entity-centric and document-centric KBs. Its integrated collective, graph-based EL algorithm is based on entity embeddings (Word2Vec) to compute an entity relatedness score and entity-context embeddings (Doc2Vec) to match the surface forms' textual context with entity descriptions. We conducted experiments on various domains and compared DoSeR to 7 strong, publicly available (table) EL systems on 16 data sets overall. DoSeR achieves state-of-the-art results over a wide range of different document types (e.g., news documents, tweets, RSS-feeds), structures (e.g., tables) and domains (e.g., general domain, biomedical domain). We showed that our approach outperforms all other systems by a significant margin on nearly all data sets. We also discussed the influence of the quality of the underlying KB on the EL accuracy and compared our results to those of other non-publicly available state-of-the-art algorithms. Further, DoSeR provides consistent results with a low quantity of existing entity data (as it is the case in the biomedical domain). We assume that DoSeR is also consistent with noisy entity data since our entity relatedness measure performs well with noisy KB data. Furthermore, if no appropriate entity descriptions are available, DoSeR achieves strong results by leveraging entity embeddings only. Experiments on how DoSeR performs with large-scale and heterogeneous KBs will conducted in the near future.

Overall, DoSeR is a robust EL framework in terms of Structural Robustness and Consistency while providing state-of-the-art results. We provide our approach as well as the underlying KB as open source solutions.

# Part V

Conclusion and Future Work

# CHAPTER 9

## Conclusion and Future Work

In this work, we focused on the research and development of a robust Entity Linking (EL) system. To this end, we defined the term *Robustness* as an umbrella term that covers two crucial characteristics of EL systems: Structural Robustness and Consistency. Structurally robust EL systems are agnostic in terms of different knowledge base (KB) types and provide consistent results across various document structures and types. In contrast, Consistency in EL systems refers to consistent results across various domains, with a low quantity and/or poor quality of entity descriptions as well as on large-scale and heterogeneous KBs. In order to create such a robust EL system, we first subdivided EL algorithms into their main components. Based on this division, we then selected the following three main components to be further investigated in terms of Structural Robustness and Consistency throughout this work: (i) the underlying KB, (ii) the entity relatedness measure, and (iii) the textual context matching technique. In the following, we briefly summarize the research questions and contributions associated with each component.

The **KB** represents the fundamental frame of an EL system and defines the underlying domain, the specific set of entities to be linked and the entity definitions that can be leveraged by an EL system. We investigated how and to which extent content-related KB properties influence EL results. More specifically, we selected and investigated the following three crucial (special-domain) KB properties: (i) the entity format, i.e., intensional and extensional entity definitions as provided by entity-centric and document-centric KBs, (ii) user data, i.e., the quantity and quality of externally disambiguated entities, and (iii) the quantity and heterogeneity of entities to disambiguate. To this end, we implemented three Learning-To-Rank-based approaches to leverage different kinds of entity definitions. The take-away message describes that a federated approach, which leverages different kinds of entity definitions, can significantly improve Consistency in EL systems.

The **entity relatedness** within an input document describes a crucial feature in collective EL algorithms. Although a plethora of entity relatedness measures has been proposed in the literature, most approaches lack Structural Robustness and/or Consistency. Hence, our research questions asked which entity relatedness measure provides Structural Robustness and Consistency while providing state-of-the-art EL results. In this context, we presented a new KB-agnostic, state-of-the-art entity relatedness measure based on semantic embeddings. In our experiments, we showed that our measure integrated in a collective, baseline EL approach outperforms other publicly available, state-of-the-art EL approaches on most data sets. Moreover, we demonstrated that our new measure is structurally robust and provides consistent results in terms of poor quality entity definitions in KBs.

The **textual context** of surface forms is also an important feature to consider in EL systems. Most textual context matching techniques integrated in existing EL systems are tailored toward specific KBs (such as Wikipedia with extensive entity descriptions) and/or specific document structures (e.g., textual documents, tables) and types (e.g., news documents, tweets). Based on these limitations, we analyzed which context matching technique provides Structural Robustness and Consistency while achieving state-of-the-art results in EL systems. We compared the neural-network-based approach Doc2Vec to four other state-of-the-art textual context matching techniques in terms of effectiveness in EL systems. More specifically, we provided a systematic evaluation of context matching techniques with regard to the document structure and type, and quantity of entity descriptions within KBs. In our experiments, we showed that Doc2Vec achieves state-of-the-art results while providing Structural Robustness. Moreover, it provides consistent results with short entity descriptions in the underlying KB.

Based on our findings and outcomes, our main contribution in this work is DoSeR (**D**isambiguation **o**f **Se**mantic **R**esources). DoSeR is a KB-agnostic EL framework that extracts relevant entity information from multiple (entity-centric and document-centric) KBs in a fully automatic way. DoSeR accepts different types of input documents such as tables, news articles and tweets, whereby each document provides one or multiple, previously annotated surface forms. The collective, graph-based EL algorithm implemented in DoSeR utilizes semantic entity (Word2Vec) and document embeddings (Doc2Vec) for entity relatedness and textual context matching computation. In our conducted experiments on general-domain and special-domain KBs, DoSeR outperformed publicly and non-publicly available, state-of-the-art EL systems on a wide range of data sets. Moreover, we showed that DoSeR achieves Structural Robustness and Consistency in terms of most criteria on the evaluated data sets and KBs. We also provide DoSeR as well as the underlying KBs as open source solutions. These resources allow a fair comparison between future EL algorithms and our approach that are not biased by the KB.

**Limitations** in our work include that our approaches and techniques were not evaluated in terms of performance. Performant EL systems are particularly important when it comes to annotating documents on a large scale. Moreover, the outcomes of our KB experiments may (slightly) differ with other EL algorithms and KBs. Nevertheless, we strongly assume that the core statements still hold. Another limitation includes that we evaluated DoSeR on a limited range of KBs. However, an additional evaluation on several other KBs is required to conclude DoSeR as a fully robust EL system.

In **future work**, we aim to significantly enrich our EL models with entity information leveraged from unstructured and/or (semi-)structured Web documents. More specifically, we will apply a state-of-the-art Entity Recognition system on a vast amount of extracted Web documents to locate surface forms. Further, we will employ DoSeR to automatically link the respective surface forms to entities from one or multiple KBs. Based on the newly acquired entity-annotated documents, we will re-train our semantic embeddings (i.e., entity-embeddings and entity-context embeddings) to incorporate knowledge from

additional information sources. With this step, we hope to further improve EL results, in particular with unpopular entities that lack descriptions in existing knowledge sources.

Moreover, we aim to conduct additional experiments where we compare DoSeR to current state-of-the-art EL systems on a wider range of specific domains and KBs. In this context, we want to provide an overview of how existing methods perform on various domains and KBs in order to investigate their weaknesses in terms of Structural Robustness and Consistency. With these additional experiments, we aim to encourage authors of future EL systems to improve the Robustness of their approaches instead of optimizing their systems on specific KBs and domains.

# Bibliography


[Ade05] Adeva, Juan Jose García, Ulises Cerviño Beresi, and Rafael A. Calvo: 'Accuracy and Diversity in Ensembles of Text Categorisers.' *CLEI Electronic Journal* (2005), vol. 8(2) (cit. on p. 62).

[Alh14a] Alhelbawy, Ayman and Robert Gaizauskas: 'Graph Ranking for Collective Named Entity Disambiguation'. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014: pp. 75–80 (cit. on p. 56).

[Alh14b] Alhelbawy, Ayman and Robert J. Gaizauskas: 'Collective Named Entity Disambiguation using Graph Ranking and Clique Partitioning Approaches'. *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*. 2014: pp. 1544–1555 (cit. on pp. 14, 15, 56, 72, 107).

[Art10] Artiles, Javier, Andrew Borthwick, Julio Gonzalo, Satoshi Sekine, and Enrique Amigó: 'WePS-3 Evaluation Campaign: Overview of the Web People Search Clustering and Attribute Extraction Tasks.' *CLEF (Notebook Papers/LABs/-Workshops)*. Ed. by Braschler, Martin, Donna Harman, and Emanuele Pianta. 2010 (cit. on p. 27).

[Aue07] Auer, Sören, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives: 'DBpedia: A Nucleus for a Web of Open Data'. *Proceedings of the 6th International Semantic Web Conference*. ISWC'07. Busan, Korea: Springer-Verlag, 2007: pp. 722–735 (cit. on pp. 3, 19).

[Bad12] Bada, Michael, Miriam Eckert, Donald Evans, Kristin Garcia, Krista Shipley, Dmitry Sitnikov, Jr Baumgartner WilliamA, KBretonnel Cohen, Karin Verspoor, JudithA Blake, and LawrenceE Hunter: 'Concept annotation in the CRAFT corpus'. English. *BMC Bioinformatics* (2012), vol. 13(1), 161 (cit. on p. 21).

[Bag98a] Bagga, Amit and Breck Baldwin: 'Algorithms for Scoring Coreference Chains'. *In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*. 1998: pp. 563–566 (cit. on p. 24).

[Bag98b] Bagga, Amit and Breck Baldwin: 'Entity-based Cross-document Coreferencing Using the Vector Space Model'. *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*. ACL '98. Montreal, Quebec, Canada: Association for Computational Linguistics, 1998: pp. 79–85 (cit. on pp. 13, 17).

[Bar14] Barrena, Ander, Eneko Agirre, Bernardo Cabaleiro, Anselmo Peñas, and Aitor Soroa: '"One Entity per Discourse" and "One Entity per Collocation" Improve Named-Entity Disambiguation'. *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*. 2014: pp. 2260–2269 (cit. on pp. 18, 72).

[Bar15] Barrena, Ander, Aitor Soroa, and Eneko Agirre: 'Combining Mention Context and Hyperlinks from Wikipedia for Named Entity Disambiguation'. *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*. Denver, Colorado: Association for Computational Linguistics, June 2015: pp. 101–105 (cit. on pp. 34, 35, 128).

[Bas14] Basave, Amparo Elizabeth Cano, Giuseppe Rizzo, Andrea Varga, Matthew Rowe, Milan Stankovic, and Aba-Sah Dadzie: 'Making Sense of Microposts Named Entity Linking Challenge'. Vol. 1141. CEUR. 2014: pp. 54–60 (cit. on p. 158).

[Ben07] Bengio, Y., P. Lamblin, D. Popovici, and H. Larochelle: 'Greedy layer-wise training of deep networks'. *Advances in Neural Information Processing Systems 19*. Ed. by Schölkopf, B., J. Platt, and T. Hoffman. Cambridge, MA: MIT Press, 2007: pp. 153–160 (cit. on p. 38).

[Ben03] Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Janvin: 'A Neural Probabilistic Language Model'. *The Journal of Machine Learning Research* (Mar. 2003), vol. 3: pp. 1137–1155 (cit. on pp. 102, 122).

[Ber01] Berners-Lee, Tim, James Hendler, and Ora Lassila: 'The Semantic Web'. *Scientific American* (May 2001), vol. 284(5): pp. 34–43 (cit. on p. 3).

[Bha15] Bhagavatula, Chandra Sekhar, Thanapon Noraset, and Doug Downey: 'TabEL: Entity Linking in Web Tables'. *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I*. 2015: pp. 425–441 (cit. on p. 40).

[Bis06] Bishop, Christopher M.: *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006 (cit. on p. 61).

[Bla15] Blanco, Roi, Giuseppe Ottaviano, and Edgar Meij: 'Fast and Space-Efficient Entity Linking for Queries'. *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. WSDM '15. Shanghai, China: ACM, 2015: pp. 179–188 (cit. on p. 35).

[Ble09] Blei, David M. and John D. Lafferty: *Topic Models*. Ed. by Srivastava, Ashok N. and Mehran Sahami. CRC Press, 2009: pp. 71–94 (cit. on p. 36).

[Ble03] Blei, David M., Andrew Y. Ng, and Michael I. Jordan: 'Latent Dirichlet Allocation'. *The Journal of Machine Learning Research* (Mar. 2003), vol. 3: pp. 993–1022 (cit. on pp. 35, 37, 122).

[Bol08] Bollacker, Kurt, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor: 'Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge'. *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. SIGMOD '08. Vancouver, Canada: ACM, 2008: pp. 1247–1250 (cit. on pp. 3, 20).

[Bos92] Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik: 'A Training Algorithm for Optimal Margin Classifiers'. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT '92. Pittsburgh, Pennsylvania, USA: ACM, 1992: pp. 144–152 (cit. on p. 61).

[Bre72] Brewer, K. R. W., L. J. Early, and S. F. Joyce: 'Selecting Several Samples from a Single Population'. *Australian Journal of Statistics* (Nov. 1972), vol. 14(3): pp. 231–239 (cit. on p. 42).

[Bri98] Brin, Sergey and Lawrence Page: 'The Anatomy of a Large-scale Hypertextual Web Search Engine'. *Proceedings of the Seventh International Conference on World Wide Web 7*. WWW7. Brisbane, Australia: Elsevier Science Publishers B. V., 1998: pp. 107–117 (cit. on pp. 32, 57, 58, 107, 108, 150).

[Bun06] Bunescu, Razvan C. and Marius Pasca: 'Using Encyclopedic Knowledge for Named entity Disambiguation'. *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*. 2006 (cit. on pp. 18, 19, 27, 34, 54, 64).

[Cai13] Cai, Zhiyuan, Kaiqi Zhao, Kenny Q. Zhu, and Haixun Wang: 'Wikification via link co-occurrence'. *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*. CIKM '13. San Francisco, California, USA: ACM, 2013: pp. 1087–1096 (cit. on p. 40).

[Cam16] Camilo Thorne, Stefano Faralli and Heiner Stuckenschmidt: 'Cross-Evaluation of Entity Linking and Disambiguation Systems for Clinical Text Annotation'. *SEMANTiCS 2016 : 12th International Conference on Semantic Systems*. New York, NY, USA: ACM, 2016: pp. 169–172 (cit. on p. 70).

[Cao07] Cao, Zhe, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li: 'Learning to Rank: From Pairwise Approach to Listwise Approach'. *Proceedings of the 24th International Conference on Machine Learning*. ICML '07. Corvalis, Oregon, USA: ACM, 2007: pp. 129–136 (cit. on p. 55).

[Cas11] Cassidy, Taylor, Zheng Chen, Javier Artiles, Heng Ji, Hongbo Deng, Lev-Arie Ratinov, Jiawei Han, Dan Roth, and Jing Zheng: 'CUNY-UIUC-SRI TAC-KBP2011 Entity Linking System Description'. *Proceedings of the Fourth Text Analysis Conference, TAC 2011, Gaithersburg, Maryland, USA, November 14-15, 2011*. 2011 (cit. on p. 62).

[Cav04] Caviedes, Jorge E. and James J. Cimino: 'Towards the Development of a Conceptual Distance Metric for the UMLS'. *Journal of Biomedical Informatics* (Apr. 2004), vol. 37(2): pp. 77–85 (cit. on p. 42).

[Cec13] Ceccarelli, Diego, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Salvatore Trani: 'Learning Relatedness Measures for Entity Linking'. *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*. CIKM '13. San Francisco, California, USA: ACM, 2013: pp. 139–148 (cit. on pp. 39, 41).

[Che05] Chen, Lifeng, Hongfang Liu, and Carol Friedman: 'Gene Name Ambiguity of Eukaryotic Nomenclatures'. *Bioinformatics* (Jan. 2005), vol. 21(2): pp. 248–256 (cit. on p. 22).

[Che06] Chen, Ping and Hisham Al-Mubaid: 'Context-based Term Disambiguation in Biomedical Literature'. *FLAIRS Conference*. AAAI Press, 2006: pp. 62–67 (cit. on pp. 34, 61).

[Che96] Chen, Stanley F. and Joshua Goodman: 'An Empirical Study of Smoothing Techniques for Language Modeling'. *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*. ACL '96. Santa Cruz, California: Association for Computational Linguistics, 1996: pp. 310–318 (cit. on p. 128).

[Che11] Chen, Zheng and Heng Ji: 'Collaborative Ranking: A Case Study on Entity Linking'. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '11. Edinburgh, United Kingdom: Association for Computational Linguistics, 2011: pp. 771–781 (cit. on pp. 34, 62).

[Che10] Chen, Zheng, Suzanne Tamang, Adam Lee, Xiang Li, Wen-Pin Lin, Matthew G. Snover, Javier Artiles, Marissa Passantino, and Heng Ji: 'Cuny-Blender TAC-KBP2010 Entity Linking and Slot Filling System Description'. *Proceedings of the TAC Workshop 2010*. 2010 (cit. on pp. 28, 29, 34, 50, 63).

[Che13] Cheng, Xiao and Dan Roth: 'Relational Inference for Wikification'. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. 2013: pp. 1787–1796 (cit. on pp. 23, 63, 111, 155).

[Chi15] Chisholm, Andrew and Ben Hachey: 'Entity Disambiguation with Web Links'. *Transactions of the Association for Computational Linguistics* (2015), vol. 3: pp. 145–156 (cit. on p. 32).

[Chr06] Christen, Peter: 'A Comparison of Personal Name Matching: Techniques and Practical Issues.' *IEEE International Conference on Data Mining Workshops*. IEEE Computer Society, 2006: pp. 290–294 (cit. on p. 31).

[Chu90] Church, Kenneth Ward and Patrick Hanks: 'Word Association Norms, Mutual Information, and Lexicography'. *Computational Linguistics* (Mar. 1990), vol. 16(1): pp. 22–29 (cit. on p. 40).

[Cil07] Cilibrasi, Rudi L. and Paul M. B. Vitanyi: 'The Google Similarity Distance'. *IEEE Transactions on Knowledge and Data Engineering* (Mar. 2007), vol. 19(3): pp. 370–383 (cit. on p. 40).

[Coh03] Cohen, William W., Pradeep Ravikumar, and Stephen E. Fienberg: 'A Comparison of String Distance Metrics for Name-Matching Tasks.' *Proceedings of IJCAI-03 Workshop on Information Integration*. Aug. 2003: pp. 73–78 (cit. on pp. 28, 31).

[Cor13] Cornolti, Marco, Paolo Ferragina, and Massimiliano Ciaramita: 'A Framework for Benchmarking Entity-annotation Systems'. *Proceedings of the 22Nd International Conference on World Wide Web*. WWW '13. Rio de Janeiro, Brazil: ACM, 2013: pp. 249–260 (cit. on pp. 22, 24).

[Cuc07] Cucerzan, Silviu: 'Large-Scale Named Entity Disambiguation Based on Wikipedia Data'. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics, June 2007: pp. 708–716 (cit. on pp. 39, 40, 50, 51, 110, 158).

[Cuc11] Cucerzan, Silviu: 'TAC Entity Linking by Performing Full-document Entity Extraction and Disambiguation'. *Proceedings of the Fourth Text Analysis Conference, TAC 2011, Gaithersburg, Maryland, USA, November 14-15, 2011*. 2011 (cit. on p. 29).

[Dai15] Dai, Andrew M., Christopher Olah, and Quoc V. Le: 'Document Embedding with Paragraph Vectors'. *CoRR* (2015), vol. abs/1507.07998 (cit. on pp. 122, 125, 137).

[Dar08] Daróczy, Bálint, Zsolt Fekete, Mátyás Brendel, Simon Rácz, András Benczúr, Dávid Siklósi, and Attila Pereszlényi: 'Cross-modal image retrieval with parameter tuning'. *CLEF*. 2008 (cit. on p. 53).

[Dav16] Davis, Allan Peter, Cynthia J. Grondin, Robin J. Johnson, Daniela Sciaky, Benjamin L. King, Roy McMorran, Jolene Wiegers, Thomas C. Wiegers, and Carolyn J. Mattingly: 'The Comparative Toxicogenomics Database: update 2017'. *Nucleic Acids Research* (2016), vol. (cit. on p. 22).

[Day08] Day, David, Janet Hitzeman, Michael Wick, Keith Crouch, and Massimo Poesio: 'A Corpus for Cross-Document Co-reference'. *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. Ed. by Calzolari, Nicoletta, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, and Daniel Tapias. Marrakech, Morocco: European Language Resources Association (ELRA), May 2008 (cit. on p. 27).

[Dem12] Demartini, Gianluca, Djellel Eddine Difallah, and Philippe Cudré-Mauroux: 'ZenCrowd: Leveraging Probabilistic Reasoning and Crowdsourcing Techniques for Large-scale Entity Linking'. *Proceedings of the 21st International Conference on World Wide Web*. WWW '12. Lyon, France: ACM, 2012: pp. 469–478 (cit. on p. 60).

[Den09] Deng, Hongbo, Irwin King, and Michael R. Lyu: 'Entropy-biased Models for Query Representation on the Click Graph'. *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '09. Boston, MA, USA: ACM, 2009: pp. 339–346 (cit. on p. 50).

[Des13] Deshpande, Omkar, Digvijay S. Lamba, Michel Tourn, Sanjib Das, Sri Subramaniam, Anand Rajaraman, Venky Harinarayan, and AnHai Doan: 'Building, Maintaining, and Using Knowledge Bases: A Report from the Trenches'. *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. SIGMOD '13. New York, NY, USA: ACM, 2013: pp. 1209–1220 (cit. on pp. 18, 20).

[Don05] Dong, Xin, Alon Halevy, and Jayant Madhavan: 'Reference Reconciliation in Complex Information Spaces'. *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*. SIGMOD '05. Baltimore, Maryland: ACM, 2005: pp. 85–96 (cit. on p. 17).

[Dre10] Dredze, Mark, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin: 'Entity Disambiguation for Knowledge Base Population'. *Proceedings of the 23rd International Conference on Computational Linguistics*. COLING '10. Beijing, China: Association for Computational Linguistics, 2010: pp. 277–285 (cit. on pp. 30–34, 54, 64).

[Elm07] Elmagarmid, Ahmed K., Panagiotis G. Ipeirotis, and Vassilios S. Verykios: 'Duplicate Record Detection: A Survey'. *IEEE Transactions on Knowledge and Data Engineering* (Jan. 2007), vol. 19(1): pp. 1–16 (cit. on p. 17).

[Fan06] Fang, Haw-ren, Kevin Murphy, Yang Jin, Jessica S. Kim, and Peter S. White: 'Human Gene Name Normalization Using Text Matching with Automatically Extracted Synonym Dictionaries'. *Proceedings of the Workshop on Linking Natural Language Processing and Biology: Towards Deeper Biological Literature Analysis*. BioNLP '06. New York City, New York: Association for Computational Linguistics, 2006: pp. 41–48 (cit. on p. 28).

[Fel98] Fellbaum, Christiane, ed.: *WordNet: an electronic lexical database*. MIT Press, 1998 (cit. on pp. 19, 155).

[Fer12] Ferragina, Paolo and Ugo Scaiella: 'Fast and Accurate Annotation of Short Texts with Wikipedia Pages'. *IEEE Software* (Jan. 2012), vol. 29(1): pp. 70–75 (cit. on p. 70).

[Fer11] Ferreira, João D. and Francisco M. Couto: 'Generic Semantic Relatedness Measure for Biomedical Ontologies'. *Proceedings of the 2nd International Conference on Biomedical Ontology, Buffalo, NY, USA, July 26-30, 2011*. 2011 (cit. on p. 42).

[Fin05] Finkel, Jenny Rose, Trond Grenager, and Christopher Manning: 'Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling'. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. ACL '05. Ann Arbor, Michigan: Association for Computational Linguistics, 2005: pp. 363–370 (cit. on p. 25).

[Fra16] Francis-Landau, Matthew, Greg Durrett, and Dan Klein: 'Capturing Semantic Similarity for Entity Linking with Convolutional Neural Networks'. *Proceedings of the North American Association for Computational Linguistics*. NAACL '16. San Diego, California, USA: Association for Computational Linguistics, June 2016 (cit. on pp. 48, 49).

[Gan16] Ganea, Octavian-Eugen, Marina Ganea, Aurélien Lucchi, Carsten Eickhoff, and Thomas Hofmann: 'Probabilistic Bag-Of-Hyperlinks Model for Entity Linking'. *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15*. 2016: pp. 927–938 (cit. on pp. 34, 35, 59, 77, 79, 128, 143, 159).

[Gat13] Gattani, Abhishek, Digvijay S. Lamba, Nikesh Garera, Mitul Tiwari, Xiaoyong Chai, Sanjib Das, Sri Subramaniam, Anand Rajaraman, Venky Harinarayan, and AnHai Doan: 'Entity Extraction, Linking, Classification, and Tagging for Social Media: A Wikipedia-based Approach'. *Proceedings of the VLDB Endowment* (Aug. 2013), vol. 6(11): pp. 1126–1137 (cit. on pp. 27, 32, 34).

[Ger13] Gerber, Daniel, Axel-Cyrille Ngonga Ngomo, Sebastian Hellmann, Tommaso Soru, Lorenz Bühmann, and Ricardo Usbeck: 'Real-time RDF extraction from unstructured data streams'. *Proceedings of the 12th International Semantic Web Conference*. 2013 (cit. on p. 110).

[Gio99] Gionis, Aristides, Piotr Indyk, and Rajeev Motwani: 'Similarity Search in High Dimensions via Hashing'. *Proceedings of the 25th International Conference on Very Large Data Bases*. VLDB '99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999: pp. 518–529 (cit. on p. 42).

[Got11] Gottipati, Swapna and Jing Jiang: 'Linking Entities to a Knowledge Base with Query Expansion'. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '11. Edinburgh, United Kingdom: Association for Computational Linguistics, 2011: pp. 804–813 (cit. on pp. 29, 52).

[Gro16] Grover, Aditya and Jure Leskovec: 'node2vec: Scalable Feature Learning for Networks'. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. 2016: pp. 855–864 (cit. on p. 115).

[Guo13] Guo, Stephen, Ming-Wei Chang, and Emre Kiciman: 'To Link or Not to Link? A Study on End-to-End Tweet Entity Linking'. *Human Language Technologies: Conference of the North American Chapter of the Association of Computational*

*Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*. 2013: pp. 1020–1030 (cit. on pp. 14, 27, 32, 34, 40, 61).

[Guo14] Guo, Zhaochen and Denilson Barbosa: 'Robust Entity Linking via Random Walks'. *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. CIKM '14. Shanghai, China: ACM, 2014: pp. 499–508 (cit. on pp. 14, 32, 41, 62, 157, 158).

[Hac13] Hachey, Ben, Will Radford, Joel Nothman, Matthew Honnibal, and James R. Curran: 'Evaluating Entity Linking with Wikipedia'. *Artificial Intelligence* (Jan. 2013), vol. 194: pp. 130–150 (cit. on pp. 13, 26).

[Had11] Hadjieleftheriou, Marios and Divesh Srivastava: 'Approximate String Processing'. *Foundations and Trends in Databases* (Apr. 2011), vol. 2(4): pp. 267–402 (cit. on p. 28).

[Hak08] Hakenberg, Jörg, Conrad Plake, Robert Leaman, Michael Schroeder, and Graciela Gonzalez: 'Inter-species normalization of gene mentions with GNAT.' *ECCB*. 2008: pp. 126–132 (cit. on p. 33).

[Han11a] Han, Xianpei and Le Sun: 'A Generative Entity-mention Model for Linking Entities with Knowledge Base'. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. HLT '11. Portland, Oregon: Association for Computational Linguistics, 2011: pp. 945–954 (cit. on pp. 32, 34, 35, 59, 64, 126, 128, 130).

[Han12] Han, Xianpei and Le Sun: 'An Entity-topic Model for Entity Linking'. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. EMNLP-CoNLL '12. Jeju Island, Korea: Association for Computational Linguistics, 2012: pp. 105–115 (cit. on pp. 46, 60, 158, 163).

[Han11b] Han, Xianpei, Le Sun, and Jun Zhao: 'Collective Entity Linking in Web Text: A Graph-based Method'. *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '11. Beijing, China: ACM, 2011: pp. 765–774 (cit. on pp. 34, 57, 107, 158).

[Han09] Han, Xianpei and Jun Zhao: 'NLPR KBP in TAC 2009 KBP Track: A Two-Stage Method to Entity Linking'. *Proceedings of the Second Text Analysis Conference, TAC 2009, Gaithersburg, Maryland, USA, November 16-17, 2009*. NIST, 2009 (cit. on pp. 30, 51).

[Han10] Han, Xianpei and Jun Zhao: 'Structural Semantic Relatedness: A Knowledge-based Method to Named Entity Disambiguation'. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. ACL '10. Uppsala, Sweden: Association for Computational Linguistics, 2010: pp. 50–59 (cit. on pp. 44, 45).

[Har12] Harmston, Nathan, Wendy Filsell, and Michael P. H. Stumpf: 'Which species is it? Species-driven gene name disambiguation using random walks over a mixture of adjacency matrices'. *Bioinformatics* (Jan. 2012), vol. 28(2): pp. 254–260 (cit. on p. 70).

[Har54] Harris, Zellig: 'Distributional structure'. *Word* (1954), vol. 10(23): pp. 146–162 (cit. on pp. 41, 122).

[Hav03] Haveliwala, Taher H.: 'Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search'. *IEEE Transactions on Knowledge and Data Engineering* (July 2003), vol. 15(4): pp. 784–796 (cit. on pp. 57, 58).

[He13a] He, Zhengyan, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang: 'Learning Entity Representation for Entity Disambiguation.' *ACL (2)*. The Association for Computer Linguistics, 2013: pp. 30–34 (cit. on pp. 38, 39).

[He13b] He, Zhengyan, Shujie Liu, Yang Song, Mu Li, Ming Zhou, and Houfeng Wang: 'Efficient Collective Entity Linking with Stacking.' *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '12. ACL, 2013: pp. 426–435 (cit. on p. 55).

[Hec12] Hecht, Brent, Samuel H. Carton, Mahmood Quaderi, Johannes Schöning, Martin Raubal, Darren Gergle, and Doug Downey: 'Explanatory Semantic Relatedness and Explicit Spatialization for Exploratory Search'. *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '12. Portland, Oregon, USA: ACM, 2012: pp. 415–424 (cit. on p. 40).

[Her00] Herbrich, R., T. Graepel, and K. Obermayer: 'Large Margin Rank Boundaries for Ordinal Regression'. *Advances in Large Margin Classifiers*. Ed. by Smola, A.J., P.L. Bartlett, B. Schölkopf, and D. Schuurmans. Cambridge, MA: MIT Press, 2000: pp. 115–132 (cit. on p. 54).

[Hin06] Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh: 'A Fast Learning Algorithm for Deep Belief Nets'. *Neural Computation* (July 2006), vol. 18(7): pp. 1527–1554 (cit. on pp. 38, 43).

[Hir05] Hirschman, L., M. Colosimo, A. Morgan, and A. Yeh: 'Overview of BioCreAtIvE task 1B: normalized gene lists'. *BMC Bioinformatics* (2005), vol. 6 (cit. on p. 16).

[Hof12] Hoffart, Johannes, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum: 'KORE: Keyphrase Overlap Relatedness for Entity Disambiguation'. *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. CIKM '12. Maui, Hawaii, USA: ACM, 2012: pp. 545–554 (cit. on pp. 34, 39, 42, 130).

[Hof11] Hoffart, Johannes, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum: 'Robust Disambiguation of Named Entities in Text'. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '11. Edinburgh, United Kingdom: Association for Computational Linguistics, 2011: pp. 782–792 (cit. on pp. 18, 32, 34, 57, 72, 79, 109, 111, 155).

[Hof13] Hoffman, Matthew D., David M. Blei, Chong Wang, and John Paisley: 'Stochastic Variational Inference'. *Journal of Machine Learning Research* (May 2013), vol. 14(1): pp. 1303–1347 (cit. on p. 36).

[Hou14] Houlsby, Neil and Massimiliano Ciaramita: 'A Scalable Gibbs Sampler for Probabilistic Entity Linking.' *ECIR*. Ed. by Rijke, Maarten de, Tom Kenter, Arjen P. de Vries, ChengXiang Zhai, Franciska de Jong, Kira Radinsky, and Katja Hofmann. Vol. 8416. Lecture Notes in Computer Science. Springer, 2014: pp. 335–346 (cit. on pp. 36, 39, 60, 77).

[Hou13] Houlsby, Neil and Massimiliano Ciaramita: 'Scalable Probabilistic Entity-Topic Modeling'. *CoRR* (2013), vol. abs/1309.0337 (cit. on p. 36).

[Hsi14] Hsiao, Jui-Chen, Chih-Hsuan Wei, and Hung-Yu Kao: 'Gene Name Disambiguation Using Multi-scope Species Detection'. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (Jan. 2014), vol. 11(1): pp. 55–62 (cit. on p. 33).

[Hsu02] Hsu, Chih-Wei and Chih-Jen Lin: 'A Comparison of Methods for Multiclass Support Vector Machines'. *IEEE Transactions on Neural Networks* (Mar. 2002), vol. 13(2): pp. 415–425 (cit. on p. 61).

[Hua14] Huang, Hongzhao, Yunbo Cao, Xiaojiang Huang, Heng Ji, and Chin-Yew Lin: 'Collective Tweet Wikification based on Semi-supervised Graph Regularization'. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014: pp. 380–390 (cit. on pp. 58, 71).

[Hua15] Huang, Hongzhao, Larry Heck, and Heng Ji: 'Leveraging Deep Neural Networks and Knowledge Graphs for Entity Disambiguation'. *CoRR* (2015), vol. abs/1504.07678 (cit. on pp. 43, 44, 72, 114, 158).

[Hua13] Huang, Po-Sen, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck: 'Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data'. *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*. CIKM '13. San Francisco, California, USA: ACM, 2013: pp. 2333–2338 (cit. on pp. 44, 45).

[Hul15] Hulpus, Ioana, Narumol Prangnawarat, and Conor Hayes: 'Path-Based Semantic Relatedness on Linked Data and Its Use to Word and Entity Disambiguation'. *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I*. 2015: pp. 442–457 (cit. on p. 43).

[Ian16] Ian Goodfellow, Yoshua Bengio and Aaron Courville: 'Deep Learning'. Book in preparation for MIT Press. 2016 (cit. on p. 43).

[Jai07] Jain, Alpa, Silviu Cucerzan, and Saliha Azzam: 'Acronym-Expansion Recognition and Ranking on the Web'. *Proceedings of the IEEE International Conference on Information Reuse and Integration, IRI 2007, 13-15 August 2007, Las Vegas, Nevada, USA*. 2007: pp. 209–214 (cit. on p. 29).

[Jel80] Jelinek, Fred and Robert L. Mercer: 'Interpolated estimation of Markov source parameters from sparse data'. *Proceedings, Workshop on Pattern Recognition in Practice*. Ed. by Gelsema, Edzard S. and Laveen N. Kanal. Amsterdam: North Holland, 1980: pp. 381–397 (cit. on pp. 35, 128).

[Ji11a] Ji, Heng and Ralph Grishman: 'Knowledge Base Population: Successful Approaches and Challenges'. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. HLT '11. Portland, Oregon: Association for Computational Linguistics, 2011: pp. 1148–1158 (cit. on pp. 16, 32, 62).

[Ji11b] Ji, Heng, Ralph Grishman, and Hoa Dang: 'Overview of the TAC2011 Knowledge Base Population Track'. *TAC 2011 Proceedings Papers*. 2011 (cit. on pp. 16, 24).

[Ji10] Ji, Heng, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis: 'Overview of the TAC 2010 knowledge base population track'. *Third Text Analysis Conference (TAC 2010)* (2010), vol. (cit. on pp. 14, 16, 33).

[Joa02] Joachims, Thorsten: 'Optimizing Search Engines Using Clickthrough Data'. *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '02. Edmonton, Alberta, Canada: ACM, 2002: pp. 133–142 (cit. on pp. 54, 91).

[Jon00] Jones, K. Sparck, S. Walker, and S. E. Robertson: 'A Probabilistic Model of Information Retrieval: Development and Comparative Experiments'. *Information Processing and Managemen* (Nov. 2000), vol. 36(6): pp. 779–808 (cit. on pp. 57, 126, 128).

[Kaf12] Kafkas, Senay, Ian Lewin, David Milward, Erik van Mulligen, Jan Kors, Udo Hahn, and Dietrich Rebholz-Schuhmann: 'CALBC: Releasing the Final Corpora'. *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey, May 2012 (cit. on pp. 20, 90).

[Kat11] Kataria, Saurabh S., Krishnan S. Kumar, Rajeev R. Rastogi, Prithviraj Sen, and Srinivasan H. Sengamedu: 'Entity Disambiguation with Hierarchical Topic Models'. *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '11. San Diego, California, USA: ACM, 2011: pp. 1037–1045 (cit. on pp. 45, 46, 60).

[Kim03] Kim, J-D, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii: 'GENIA Corpus — A Semantically Annotated Corpus for Bio-Textmining'. *Bioinformatics* (2003), vol. 19(suppl 1): pp. 180–182 (cit. on p. 21).

[Kir15] Kiros, Ryan, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler: 'Skip-Thought Vectors'. *Advances in Neural Information Processing Systems 28*. Ed. by Cortes, C., N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, Inc., 2015: pp. 3294–3302 (cit. on p. 122).

[Kle99] Kleinberg, Jon M.: 'Authoritative Sources in a Hyperlinked Environment'. *Journal of the ACM* (Sept. 1999), vol. 46(5): pp. 604–632 (cit. on pp. 56, 57).

[Kol09] Koller, Daphne and Nir Friedman: *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009 (cit. on p. 60).

[Kot00] Kotz, Samuel, Narayanaswamy Balakrishnan, and Norman Lloyd Johnson: *Continuous multivariate distributions. Volume 1. , Models and applications*. Wiley series in probability and statistics. New York, Chichester, Weinheim: J. Wiley & sons, 2000 (cit. on p. 36).

[Ksc01] Kschischang, Frank R., Brendan J. Frey, and Hans-Andrea Loeliger: 'Factor graphs and the sum-product algorithm'. *IEEE Transaction on Information Theory* (2001), vol. 47(2): pp. 498–519 (cit. on p. 60).

[Kul09] Kulkarni, Sayali, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti: 'Collective Annotation of Wikipedia Entities in Web Text'. *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '09. Paris, France: ACM, 2009: pp. 457–466 (cit. on pp. 34, 46, 54, 60, 64, 110).

[Kul51] Kullback, S. and R. A. Leibler: 'On Information and Sufficiency'. *Annals of Mathematical Statistics* (1951), vol. 22(1): pp. 79–86 (cit. on pp. 36, 41, 52, 62, 129).

[Kum11] Kumar, Abhimanu and Matthew Lease: 'Learning to Rank from a Noisy Crowd'. *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '11. Beijing, China: ACM, 2011: pp. 1221–1222 (cit. on p. 98).

[Kus15] Kusner, Matt J., Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger: 'From Word Embeddings To Document Distances'. *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. 2015: pp. 957–966 (cit. on p. 122).

[Lao10] Lao, Ni and William W. Cohen: 'Relational Retrieval Using a Combination of Path-constrained Random Walks'. *Journal of Machine Learning Research* (Oct. 2010), vol. 81(1): pp. 53–67 (cit. on p. 60).

[Lau16] Lau, Jey Han and Timothy Baldwin: 'An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation'. *CoRR* (2016), vol. abs/1607.05368 (cit. on pp. 122, 125, 130, 137).

[Le14] Le, Quoc V. and Tomas Mikolov: 'Distributed Representations of Sentences and Documents'. *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*. 2014: pp. 1188–1196 (cit. on pp. 122–124, 137).

[LeC98] LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner: 'Gradient-Based Learning Applied to Document Recognition'. *Proceedings of the IEEE*. Vol. 86. 11. 1998: pp. 2278–2324 (cit. on p. 48).

[Leh10] Lehmann, John, Sean Monahan, Luke Nezda, Arnold Jung, and Ying Shi: 'LCC Approaches to Knowledge Base Population at TAC 2010'. *Proceedings of the Third Text Analysis Conference, TAC 2010, Gaithersburg, Maryland, USA, November 15-16, 2010*. 2010 (cit. on pp. 29–31, 33).

[Lei05] Leicht, E. A., Petter Holme, and M. E. J. Newman: *Vertex similarity in networks*. cite arxiv:physics/0510143. 2005 (cit. on p. 44).

[Li11] Li, Hang: 'A Short Introduction to Learning to Rank.' *IEICE Transactions* (2011), vol. 94-D(10): pp. 1854–1862 (cit. on p. 54).

[Li06] Li, Wei and Andrew McCallum: 'Pachinko Allocation: DAG-structured Mixture Models of Topic Correlations'. *Proceedings of the 23rd International Conference on Machine Learning*. ICML '06. Pittsburgh, Pennsylvania, USA: ACM, 2006: pp. 577–584 (cit. on p. 45).

[Li16] Li, Yang, Shulong Tan, Huan Sun, Jiawei Han, Dan Roth, and Xifeng Yan: 'Entity Disambiguation with Linkless Knowledge Bases'. *Proceedings of the 25th International Conference on World Wide Web*. WWW '16. Montreal, Quebec, Canada: International World Wide Web Conferences Steering Committee, 2016: pp. 1261–1270 (cit. on pp. 47, 48, 60).

[Li13] Li, Yang, Chi Wang, Fangqiu Han, Jiawei Han, Dan Roth, and Xifeng Yan: 'Mining Evidences for Named Entity Disambiguation'. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '13. Chicago, Illinois, USA: ACM, 2013: pp. 1070–1078 (cit. on pp. 37, 47, 64).

[Lim10] Limaye, Girija, Sunita Sarawagi, and Soumen Chakrabarti: 'Annotating and Searching Web Tables Using Entities, Types and Relationships'. *Proceedings of the VLDB Endowment* (Sept. 2010), vol. 3(1-2): pp. 1338–1347 (cit. on pp. 42, 60, 130, 152, 159, 160).

[Lin98] Lin, Dekang: 'An Information-Theoretic Definition of Similarity'. *Proceedings of the Fifteenth International Conference on Machine Learning*. ICML '98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998: pp. 296–304 (cit. on pp. 41, 44).

[Lin06] Lin, J.: 'Divergence Measures Based on the Shannon Entropy'. *IEEE Transactions on Information Theory* (Sept. 2006), vol. 37(1): pp. 145–151 (cit. on p. 36).

[Lin15] Ling, Xiao, Sameer Singh, and Daniel S. Weld: 'Design Challenges for Entity Linking'. *Transactions of the Association for Computational Linguistics* (2015), vol. 3: pp. 315–328 (cit. on pp. 14, 65).

[Liu09] Liu, Tie-Yan: 'Learning to Rank for Information Retrieval'. *Foundations and Trends in Information Retrieval* (Mar. 2009), vol. 3(3): pp. 225–331 (cit. on pp. 41, 54).

[Liu13] Liu, Xiaohua, Yitong Li, Haocheng Wu, Ming Zhou, Furu Wei, and Yi Lu: 'Entity Linking for Tweets.' *ACL (1)*. The Association for Computer Linguistics, 2013: pp. 1304–1311 (cit. on pp. 31, 32, 34, 36).

[Lu11] Lu, Zhiyong et al.: 'The gene normalization task in BioCreative III'. *BMC Bioinformatics* (2011), vol. 12(8): pp. 1–19 (cit. on p. 16).

[Mag11] Magrane, Michele: 'UniProt Knowledgebase: a hub of integrated protein data'. *Database* (2011), vol. 2011 (cit. on p. 22).

[Mah15] Mahdisoltani, Farzaneh, Joanna Biega, and Fabian M. Suchanek: 'YAGO3: A Knowledge Base from Multilingual Wikipedias'. *Seventh Biennial Conference on Innovative Data Systems Research*. CIDR'15. Asilomar, CA, USA: Online Proceedings, 2015 (cit. on p. 20).

[Man08] Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze: *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008 (cit. on pp. 23, 34, 50, 88, 91, 128).

[McC03] McCarthy, Diana and John Carroll: 'Disambiguating Nouns, Verbs, and Adjectives Using Automatically Acquired Selectional Preferences'. *Computational Linguistics* (Dec. 2003), vol. 29(4): pp. 639–654 (cit. on p. 16).

[McN09] McNamee, Paul and Hoa T. Dang: 'Overview of the TAC 2009 knowledge base population track'. *In Proceedings of the 2009 Text Analysis Conference*. National Institute of Standards and Technology, Nov. 2009 (cit. on pp. 15, 16).

[Men11] Mendes, Pablo N., Max Jakob, Andres Garcia-Silva, and Christian Bizer: 'DBpedia Spotlight: Shedding Light on the Web of Documents'. *Proceedings of the 7th International Conference on Semantic Systems*. I-Semantics '11. Graz, Austria: ACM, 2011: pp. 1–8 (cit. on pp. 50, 110, 111).

[Mih07] Mihalcea, Rada and Andras Csomai: 'Wikify!: Linking Documents to Encyclopedic Knowledge'. *Proceedings of the 16th ACM Conference on Information and Knowledge Management*. CIKM '07. Lisbon, Portugal: ACM, 2007: pp. 233–242 (cit. on p. 15).

[Mik13a] Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean: 'Efficient Estimation of Word Representations in Vector Space'. *CoRR* (2013), vol. abs/1301.3781 (cit. on pp. 35, 48, 101, 102, 111, 114, 122, 124, 154, 164).

[Mik13b] Mikolov, Tomas, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean: 'Distributed Representations of Words and Phrases and their Compositionality'. *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. December 5-8, 2013, Lake Tahoe, Nevada, United States.* 2013: pp. 3111–3119 (cit. on pp. 101–103).

[Mil08a] Milne, David and Ian H. Witten: 'An effective, low-cost measure of semantic relatedness obtained from Wikipedia links'. *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy.* AAAI Press, July 2008: pp. 25–30 (cit. on pp. 40, 44, 45, 51, 55, 57, 58).

[Mil13] Milne, David and Ian H. Witten: 'An Open-source Toolkit for Mining Wikipedia'. *Artificial Intelligence* (Jan. 2013), vol. 194: pp. 222–239 (cit. on p. 40).

[Mil08b] Milne, David and Ian H. Witten: 'Learning to Link with Wikipedia'. *Proceedings of the 17th ACM Conference on Information and Knowledge Management.* CIKM '08. Napa Valley, California, USA: ACM, 2008: pp. 509–518 (cit. on pp. 23, 40, 51, 110).

[Mon11] Monahan, Sean, John Lehmann, Timothy Nyberg, Jesse Plymale, and Arnold Jung: 'Cross-Lingual Cross-Document Coreference with Entity Linking'. *Proceedings of the Fourth Text Analysis Conference, TAC 2011, Gaithersburg, Maryland, USA, November 14-15, 2011.* 2011 (cit. on pp. 30, 31, 33, 61).

[Mor08a] Moreau, Erwan, François Yvon, and Olivier Cappé: 'Robust Similarity Measures for Named Entities Matching'. *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1.* COLING '08. Manchester, United Kingdom: Association for Computational Linguistics, 2008: pp. 593–600 (cit. on p. 28).

[Mor08b] Morgan, Alexander A et al.: 'Overview of BioCreative II gene normalization'. English. *Genome Biology* (2008), vol. 9(Suppl 2) (cit. on pp. 15, 16).

[Mor14] Moro, Andrea, Alessandro Raganato, and Roberto Navigli: 'Entity Linking meets Word Sense Disambiguation: a Unified Approach'. *Transactions of the Association for Computational Linguistics* (2014), vol. 2: pp. 231–244 (cit. on pp. 13, 16, 57, 62, 70, 111, 155).

[Mul13] Mulwad, Varish, Tim Finin, and Anupam Joshi: 'Semantic Message Passing for Generating Linked Data from Tables.' *International Semantic Web Conference (1).* Ed. by Alani, Harith, Lalana Kagal, Achille Fokoue, Paul T. Groth, Chris Biemann, Josiane Xavier Parreira, Lora Aroyo, Natasha F. Noy, Chris Welty, and Krzysztof Janowicz. Vol. 8218. Lecture Notes in Computer Science. Springer, 2013: pp. 363–378 (cit. on pp. 60, 159).

[Mur99] Murphy, Kevin P., Yair Weiss, and Michael I. Jordan: 'Loopy Belief Propagation for Approximate Inference: An Empirical Study'. *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence.* UAI'99. Stockholm, Sweden: Morgan Kaufmann Publishers Inc., 1999: pp. 467–475 (cit. on p. 59).

[Nad07] Nadeau, David and Satoshi Sekine: 'A survey of named entity recognition and classification'. *Linguisticae Investigationes* (Jan. 2007), vol. 30(1). Publisher: John Benjamins Publishing Company: pp. 3–26 (cit. on p. 25).

[Nav09] Navigli, Roberto: 'Word Sense Disambiguation: A Survey'. *ACM Computing Surveys* (Feb. 2009), vol. 41(2): 10:1–10:69 (cit. on pp. 16, 17).

[Nav12] Navigli, Roberto and Simone Paolo Ponzetto: 'BabelNet: The Automatic Construction, Evaluation and Application of a Wide-coverage Multilingual Semantic Network'. *Journal of Artificial Intelligence* (Dec. 2012), vol. 193: pp. 217–250 (cit. on p. 57).

[Nee70] Needleman, Saul B. and Christian D. Wunsch: 'A general method applicable to the search for similarities in the amino acid sequence of two proteins'. *Journal of Molecular Biology* (Mar. 1970), vol. 48(3): pp. 443–453 (cit. on p. 28).

[Nem10] Nemeskey, Dávid, Gábor Recski, Attila Zséder, and András Kornai: 'Budapestacad at TAC 2010'. *Proceedings of the Third Text Analysis Conference, TAC 2010, Gaithersburg, Maryland, USA, November 15-16, 2010*. 2010 (cit. on pp. 33, 53, 63).

[Ogd23] Ogden, C.K. and I. A. Richards: 'The Meaning of Meaning: A Study of the Influence of Language Upon Thought and of the Science of Symbolism.' *8th ed. 1923. Reprint New York: Harcourt Brace Jovanovich* (1923), vol. (cit. on pp. 17, 77, 82).

[Opi99] Opitz, David W. and Richard Maclin: 'Popular Ensemble Methods: An Empirical Study.' *Journal of Artificial Intelligence Research* (1999), vol. 11: pp. 169–198 (cit. on p. 62).

[Pas14] Passos, Alexandre, Vineet Kumar, and Andrew McCallum: 'Lexicon Infused Phrase Embeddings for Named Entity Resolution'. *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*. Ann Arbor, Michigan: Association for Computational Linguistics, June 2014: pp. 78–86 (cit. on p. 25).

[Pen14] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning: 'Glove: Global Vectors for Word Representation'. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. 2014: pp. 1532–1543 (cit. on pp. 102, 122).

[Per14] Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena: 'DeepWalk: Online Learning of Social Representations'. *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '14. New York, New York, USA: ACM, 2014: pp. 701–710 (cit. on p. 114).

[Pic14] Piccinno, Francesco and Paolo Ferragina: 'From TagME to WAT: A New Entity Annotator'. *First Int. Workshop on Entity Recognition/Disambiguation*. ERD '14. Gold Coast, Queensland, Australia: ACM, 2014: pp. 55–62 (cit. on pp. 57, 107, 111, 155).

[Pil11] Pilz, Anja and Gerhard Paaß: 'From Names to Entities Using Thematic Context Distance'. *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. CIKM '11. Glasgow, Scotland, UK: ACM, 2011: pp. 857–866 (cit. on pp. 36, 39, 61, 126, 129, 130).

[Poe11] Poesio, M., S. Ponzetto, and Y. Versley: 'Computational models of anaphora resolution: A survey'. *Linguistic Issues in Language Technology* (2011), vol. (cit. on p. 17).

[Pon07] Ponzetto, Simone Paolo and Michael Strube: 'Knowledge Derived from Wikipedia for Computing Semantic Relatedness'. *Journal of Artificial Intelligence Research* (2007), vol. 30: pp. 181–212 (cit. on p. 40).

[Por08] Porteous, Ian, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling: 'Fast Collapsed Gibbs Sampling for Latent Dirichlet Allocation'. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '08. Las Vegas, Nevada, USA: ACM, 2008: pp. 569–577 (cit. on p. 36).

[Rad89] Rada, R., H. Mili, E. Bicknell, and M. Blettner: 'Development and application of a metric on semantic nets'. *IEEE Transactions on Systems, Man, and Cybernetics* (Jan. 1989), vol. 19(1): pp. 17–30 (cit. on p. 42).

[Raj11] Rajaraman, Anand and Jeffrey David Ullman: *Mining of Massive Datasets*. New York, NY, USA: Cambridge University Press, 2011 (cit. on p. 127).

[Rao13] Rao, Delip, Paul McNamee, and Mark Dredze: 'Multi-source, Multilingual Information Extraction and Summarization'. Ed. by Poibeau, Thierry, Horacio Saggion, Jakub Piskorski, and Roman Yangarber. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. Chap. Entity Linking: Finding Extracted Entities in a Knowledge Base (cit. on p. 17).

[Rat09] Ratinov, Lev and Dan Roth: 'Design Challenges and Misconceptions in Named Entity Recognition'. *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*. CoNLL '09. Boulder, Colorado: Association for Computational Linguistics, 2009: pp. 147–155 (cit. on p. 25).

[Rat11] Ratinov, Lev, Dan Roth, Doug Downey, and Mike Anderson: 'Local and Global Algorithms for Disambiguation to Wikipedia'. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. HLT '11. Portland, Oregon: Association for Computational Linguistics, 2011: pp. 1375–1384 (cit. on pp. 23, 24, 31, 32, 34, 40, 54, 55, 62, 64, 79, 109, 111, 143, 155, 158).

[Reb10] Rebholz-Schuhmann, Dietrich, Antonio José Jimeno Yepes, Erik M Van Mulligen, Jan Kors, David Milward, Peter Corbett, Ekaterina Buyko, Elena Beisswanger, and Udo Hahn: 'CALBC silver standard corpus'. *Journal of Bioinformatics and Computational Biology* (2010), vol. 8(01): pp. 163–179 (cit. on p. 20).

[Řeh10] Řehůřek, Radim and Petr Sojka: 'Software Framework for Topic Modelling with Large Corpora'. English. *Proc of the LREC 2010 Workshop*. Valletta, Malta: ELRA, May 2010: pp. 45–50 (cit. on pp. 104, 111, 124, 125, 129, 153).

[Res95] Resnik, Philip: 'Using Information Content to Evaluate Semantic Similarity in a Taxonomy'. *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*. IJCAI'95. Montreal, Quebec, Canada: Morgan Kaufmann Publishers Inc., 1995: pp. 448–453 (cit. on pp. 32, 88).

[Röd14] Röder, Michael, Ricardo Usbeck, Sebastian Hellmann, Daniel Gerber, and Andreas Both: 'N3 - A Collection of Datasets for Named Entity Recognition and Disambiguation in the NLP Interchange Format'. *The 9th edition of the Language Resources and Evaluation Conference, 26-31 May, Reykjavik, Iceland*. 2014 (cit. on p. 110).

[Rud14] Rudniy, Alex, Min Song, and James Geller: 'Mapping biological entities using the Longest Approximately Common Prefix method'. *BMC Bioinformatics* (2014), vol. 15: p. 187 (cit. on p. 28).

[Ryb14] Rybinski, Maciej and José Francisco Aldana-Montes: 'Calculating Semantic Relatedness for Biomedical Use in a Knowledge-poor Environment'. *BMC Bioinformatics* (2014), vol. 15(14): pp. 1–16 (cit. on pp. 34, 42).

[Sal88] Salton, Gerard and Christopher Buckley: 'Term-weighting Approaches in Automatic Text Retrieval'. *Information Processing and Management* (Aug. 1988), vol. 24(5): pp. 513–523 (cit. on pp. 28, 31, 126, 127).

[Sen12] Sen, Prithviraj: 'Collective Context-aware Topic Models for Entity Disambiguation'. *Proceedings of the 21st International Conference on World Wide Web*. WWW '12. Lyon, France: ACM, 2012: pp. 729–738 (cit. on pp. 46, 77).

[Sha51] Shannon, Claude Elwood: 'Prediction and Entropy of Printed English'. *Bell System Technical Journal* (Jan. 1951), vol. 30: pp. 50–64 (cit. on p. 50).

[She05] Shen, Libin and Aravind K. Joshi: 'Ranking and Reranking with Perceptron'. *Machine Learning* (Sept. 2005), vol. 60(1-3): pp. 73–96 (cit. on p. 55).

[She14] Shen, Wei, Jiawei Han, and Jianyong Wang: 'A Probabilistic Model for Linking Named Entities in Web Text with Heterogeneous Information Networks'. *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*. SIGMOD '14. Snowbird, Utah, USA: ACM, 2014: pp. 1199–1210 (cit. on pp. 32, 59, 71).

[She15] Shen, Wei, Jianyong Wang, and Jiawei Han: 'Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions'. *Transactions on Knowledge & Data Engineering* (2015), vol. 27(2): pp. 443–460 (cit. on pp. 14, 17, 27, 28, 30, 31, 34, 54, 64, 65).

[She12a] Shen, Wei, Jianyong Wang, Ping Luo, and Min Wang: 'LIEGE: Link Entities in Web Lists with Knowledge Base'. *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '12. Beijing, China: ACM, 2012: pp. 1424–1432 (cit. on pp. 41, 54, 55).

[She12b] Shen, Wei, Jianyong Wang, Ping Luo, and Min Wang: 'LINDEN: linking named entities with knowledge base via semantic knowledge'. *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*. 2012: pp. 449–458 (cit. on pp. 28, 32, 54, 55, 70).

[She13] Shen, Wei, Jianyong Wang, Ping Luo, and Min Wang: 'Linking Named Entities in Tweets with Knowledge Base via User Interest Modeling'. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '13. Chicago, Illinois, USA: ACM, 2013: pp. 68–76 (cit. on pp. 19, 34, 57).

[Sil13] Sil, Avirup and Alexander Yates: 'Re-ranking for joint named-entity recognition and linking'. *Proceedings of the 22nd ACM international conference on Information and Knowledge Management*. CIKM '13. San Francisco, California, USA: ACM, 2013: pp. 2369–2374 (cit. on p. 14).

[Smi03] Smith, Lawrence H., Lana Yeganova, and W. John Wilbur: 'Hidden Markov models and optimized sequence alignments'. *Computational Biology and Chemistry* (2003), vol. 27(1): pp. 77–84 (cit. on p. 28).

[Soz10] Sozio, Mauro and Aristides Gionis: 'The Community-search Problem and How to Plan a Successful Cocktail Party'. *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '10. Washington, DC, USA: ACM, 2010: pp. 939–948 (cit. on p. 57).

[Suc07] Suchanek, Fabian M., Gjergji Kasneci, and Gerhard Weikum: 'Yago: A Core of Semantic Knowledge'. *Proceedings of the 16th International Conference on World Wide Web*. WWW '07. Banff, Alberta, Canada: ACM, 2007: pp. 697–706 (cit. on pp. 3, 19).

[Tan15] Tang, Jian, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei: 'LINE: Large-scale Information Network Embedding'. *Proceedings of the 24th International Conference on World Wide Web*. WWW '15. Florence, Italy: ACM, 2015: pp. 1067–1077 (cit. on p. 114).

[Tap05] Tapio Pahikkala Filip Ginter, Jorma Boberg: 'Contextual weighting for Support Vector Machines in literature mining: an application to gene versus protein name disambiguation'. *BMC Bioinformatics* (2005), vol. (1): p. 157 (cit. on p. 70).

[Tia13] Tian, Li, Weinan Zhang, Antonis Bikakis, Haofen Wang, Yong Yu, Yuan Ni, and Feng Cao: 'MeDetect: A LOD-Based System for Collective Entity Annotation in Biomedicine'. *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 01*. WI-IAT '13. Washington, DC, USA: IEEE Computer Society, 2013: pp. 233–240 (cit. on p. 70).

[Ton06] Tong, Hanghang, Christos Faloutsos, and Jia-Yu Pan: 'Fast Random Walk with Restart and Its Applications'. *Proceedings of the Sixth International Conference on Data Mining*. ICDM '06. Washington, DC, USA: IEEE Computer Society, 2006: pp. 613–622 (cit. on pp. 41, 62).

[Tso05] Tsochantaridis, Ioannis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun: 'Large Margin Methods for Structured and Interdependent Output Variables'. *The Journal of Machine Learning Research* (Dec. 2005), vol. 6: pp. 1453–1484 (cit. on p. 61).

[Tsu07] Tsuruoka, Yoshimasa, John McNaught, Jun'ichi Tsujii, and Sophia Ananiadou: 'Learning string similarity measures for gene/protein name dictionary look-up using logistic regression.' *Bioinformatics* (2007), vol. 23(20): pp. 2768–2774 (cit. on pp. 27, 28).

[Usb14] Usbeck, Ricardo, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, SandroAthaide Coelho, Sören Auer, and Andreas Both: 'AGDISTIS - Graph-Based Disambiguation of Named Entities Using Linked Data'. English. *The Semantic Web – ISWC 2014*. Ed. by Mika, Peter, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig Knoblock, Denny Vrandečić, Paul Groth, Natasha Noy, Krzysztof Janowicz, and Carole Goble. Vol. 8796. Lecture Notes in Computer Science. Springer International Publishing, 2014: pp. 457–471 (cit. on pp. 4, 28, 42, 56, 70, 71, 77, 79, 106, 108, 111).

[Usb15] Usbeck, Ricardo et al.: 'GERBIL: General Entity Annotator Benchmarking Framework'. *Proceedings of the 24th International Conference on World Wide Web*. WWW '15. Florence, Italy: ACM, 2015: pp. 1133–1143 (cit. on pp. 22, 24, 109–111, 130, 149).

[Var07] Varga, Dániel and Eszter Simon: 'Hungarian Named Entity Recognition with a Maximum Entropy Approach'. *Acta Cybernetica* (Feb. 2007), vol. 18(2): pp. 293–301 (cit. on p. 33).

[Var10] Varma, Vasudeva, Praveen Bysani, Kranthi Reddy B, Vijay Bharath Reddy, Sudheer Kovelamudi, Srikanth Reddy Vaddepally, Radheshyam Nanduri, N. Kiran Kumar, Santhosh Gsk, and Prasad Pingali: 'IIIT Hyderabad in Guided Summarization and Knowledge Base Population'. *Proceedings of the Third Text Analysis Conference, TAC 2010, Gaithersburg, Maryland, USA, November 15-16, 2010*. 2010 (cit. on pp. 29, 53).

[Var09] Varma, Vasudeva, Vijay Bharath Reddy, Sudheer Kovelamudi, Praveen Bysani, Santhosh Gsk, N. Kiran Kumar, Kranthi Reddy B, Karuna Kumar, and Nitin Maganti: 'IIIT Hyderabad at TAC 2009'. *Proceedings of the Second Text Analysis Conference, TAC 2009, Gaithersburg, Maryland, USA, November 16-17, 2009*. 2009 (cit. on pp. 53, 61, 63).

[Wan15] Wang, Han, Jinguang Zheng, Xiaogang Ma, Peter Fox, and Heng Ji: 'Language and Domain Independent Entity Linking with Quantified Collective Validation.' *EMNLP*. Ed. by Màrquez, Lluís, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton. The Association for Computational Linguistics, 2015: pp. 695–704 (cit. on p. 70).

[Wan09] Wang, Xinglong, Jun'ichi Tsujii, and Sophia Ananiadou: 'Classifying Relations for Biomedical Named Entity Disambiguation'. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*. EMNLP '09. Singapore: Association for Computational Linguistics, 2009: pp. 1513–1522 (cit. on p. 33).

[Wan10] Wang, Xinglong, Jun'ichi Tsujii, and Sophia Ananiadou: 'Disambiguating the species of biomedical named entities using natural language parsers'. *Bioinformatics* (2010), vol. 26(5): pp. 661–667 (cit. on p. 33).

[Whi03] White, Scott and Padhraic Smyth: 'Algorithms for Estimating Relative Importance in Networks'. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '03. Washington, D.C.: ACM, 2003: pp. 266–275 (cit. on pp. 107–109, 150, 151, 154).

[Win90] Winkler, William E.: 'String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage'. *Proceedings of the Section on Survey Research*. Wachington, DC, 1990: pp. 354–359 (cit. on pp. 28, 86, 88).

[Wol92] Wolpert, David H.: 'Original Contribution: Stacked Generalization'. *Neural Networks* (Feb. 1992), vol. 5(2): pp. 241–259 (cit. on p. 55).

[Wu12] Wu, Wentao, Hongsong Li, Haixun Wang, and Kenny Q. Zhu: 'Probase: A Probabilistic Taxonomy for Text Understanding'. *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. SIGMOD '12. Scottsdale, Arizona, USA: ACM, 2012: pp. 481–492 (cit. on pp. 3, 20).

[Yak10] Yakout, Mohamed, Ahmed K. Elmagarmid, Hazem Elmeleegy, Mourad Ouzzani, and Alan Qi: 'Behavior Based Record Linkage'. *Proceedings of the VLDB Endowment* (Sept. 2010), vol. 3(1-2): pp. 439–448 (cit. on p. 17).

[Zha04] Zhai, Chengxiang and John Lafferty: 'A Study of Smoothing Methods for Language Models Applied to Information Retrieval'. *ACM Transactions on Information Systems* (Apr. 2004), vol. 22(2): pp. 179–214 (cit. on p. 52).

[Zha01] Zhai, Chengxiang and John Lafferty: 'Model-based Feedback in the Language Modeling Approach to Information Retrieval'. *Proceedings of the Tenth International Conference on Information and Knowledge Management*. CIKM '01. Atlanta, Georgia, USA: ACM, 2001: pp. 403–410 (cit. on p. 52).

[Zha13a] Zhang, Tao, Kang Liu, and Jun Zhao: 'Cross Lingual Entity Linking with Bilingual Topic Model'. *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. IJCAI '13. Beijing, China: AAAI Press, 2013: pp. 2218–2224 (cit. on p. 15).

[Zha11a] Zhang, Wei, Yan Chuan Sim, Jian Su, and Chew Lim Tan: 'Entity Linking with Effective Acronym Expansion, Instance Selection and Topic Modeling'. *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*. IJCAI'11. Barcelona, Catalonia, Spain: AAAI Press, 2011: pp. 1909–1914 (cit. on pp. 29, 36, 54, 64).

[Zha11b] Zhang, Wei, Jian Su, Bin Chen, Wenting Wang, Zhiqiang Toh, Yanchuan Sim, Yunbo Cao, Chin Yew Lin, and Chew Lim Tan: 'I2R-NUS-MSRA at TAC 2011: Entity Linking'. *Proceedings of the TAC Workshop*. 2011 (cit. on p. 54).

[Zha10a] Zhang, Wei, Jian Su, Chew Lim Tan, and Wen Ting Wang: 'Entity Linking Leveraging Automatically Generated Annotation'. *Proceedings of the 23rd International Conference on Computational Linguistics*. COLING '10. Beijing, China: Association for Computational Linguistics, 2010: pp. 1290–1298 (cit. on pp. 28, 34, 61).

[Zha10b] Zhang, Wei, Chew Lim Tan, Yan Chuan Sim, and Jian Su: 'NUS-I2R: Learning a Combined System for Entity Linking'. *Proceedings of the Third Text Analysis Conference, TAC 2010, Gaithersburg, Maryland, USA, November 15-16, 2010*. 2010 (cit. on pp. 29, 53, 61, 62).

[Zha16] Zhang, Ziqi: 'Effective and Efficient Semantic Table Interpretation using TableMiner+'. *Semantic Web, to appear* (2016), vol. 7 (cit. on pp. 159, 160).

[Zha14] Zhang, Ziqi: 'Towards Efficient and Effective Semantic Table Interpretation'. *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*. 2014: pp. 487–502 (cit. on p. 152).

[Zha13b] Zhang, Ziqi, Anna Lisa Gentile, and Fabio Ciravegna: 'Recent advances in methods of lexical semantic relatedness - a survey.' *Natural Language Engineering* (2013), vol. 19(4): pp. 411–479 (cit. on p. 40).

[Zhe14] Zheng, Jin Guang, Daniel Howsmon, Boliang Zhang, Juergen Hahn, Deborah McGuinness, James Hendler, and Heng Ji: 'Entity Linking for Biomedical Literature'. *Proceedings of the ACM 8th International Workshop on Data and Text Mining in Bioinformatics*. DTMBIO '14. Shanghai, China: ACM, 2014: pp. 3–4 (cit. on p. 58).

[Zhe10] Zheng, Zhicheng, Fangtao Li, Minlie Huang, and Xiaoyan Zhu: 'Learning to Link Entities with Knowledge Base'. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. HLT '10. Los Angeles, California: Association for Computational Linguistics, 2010: pp. 483–491 (cit. on pp. 28, 29, 31, 55, 64).

[Zhu03] Zhu, Xiaojin, Zoubin Ghahramani, and John Lafferty: 'Semi-supervised learning using Gaussian fields and harmonic functions'. *Proceedings of the Fifteenth International Conference on Machine Learning*. ICML'03. 2003: pp. 912–919 (cit. on p. 58).

[Zwi13a] Zwicklbauer, Stefan, Christoph Einsiedler, Michael Granitzer, and Christin Seifert: 'Towards Disambiguating Web Tables'. *Proceedings of the ISWC 2013 Posters & Demonstrations Track, Sydney, Australia, October 23, 2013*. 2013: pp. 205–208 (cit. on pp. 10, 159).

[Zwi13b] Zwicklbauer, Stefan, Christin Seifert, and Michael Granitzer: 'Do We Need Entity-Centric Knowledge Bases for Entity Disambiguation?' *13th International Conference on Knowledge Management and Knowledge Technologies, I-KNOW '13, Graz, Austria, September 4-6, 2013*. 2013: 4:1–4:8 (cit. on pp. 6, 9, 81, 163).

[Zwi16a] Zwicklbauer, Stefan, Christin Seifert, and Michael Granitzer: 'DoSeR - A Knowledge-Base-Agnostic Framework for Entity Disambiguation Using Semantic Embeddings'. *The Semantic Web. Latest Advances and New Domains - 13th International Conference, ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Proceedings*. 2016: pp. 182–198 (cit. on pp. 6, 10, 101, 143).

[Zwi15a] Zwicklbauer, Stefan, Christin Seifert, and Michael Granitzer: 'From General to Specialized Domain: Analyzing Three Crucial Problems of Biomedical Entity Disambiguation'. *Database and Expert Systems Applications - 26th International Conference, DEXA 2015, Valencia, Spain, September 1-4, 2015, Proceedings, Part I*. 2015: pp. 76–93 (cit. on pp. 6, 9, 17, 18, 70, 76, 81).

[Zwi15b] Zwicklbauer, Stefan, Christin Seifert, and Michael Granitzer: 'Linking Biomedical Data to the Cloud'. *Smart Health - Open Problems and Future Challenges*. 2015: pp. 209–235 (cit. on pp. 10, 13, 14, 20, 21, 33, 65, 70, 82).

[Zwi16b] Zwicklbauer, Stefan, Christin Seifert, and Michael Granitzer: 'Robust and Collective Entity Disambiguation Through Semantic Embeddings'. *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '16. Pisa, Italy: ACM, 2016: pp. 425–434 (cit. on pp. 6, 10, 130, 137, 138, 143).

[Zwi15c] Zwicklbauer, Stefan, Christin Seifert, and Michael Granitzer: 'Search-based Entity Disambiguation with Document-centric Knowledge Bases'. *Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business, I-KNOW '15, Graz, Austria, October 21-23, 2015*. 2015: 6:1–6:8 (cit. on pp. 6, 9, 81).

# List of Figures

# List of Tables