From unstructured to structured: Context-based Named Entity Mining from Text

Zhe Zuo
Hasso-Platter-Institut Potsdam
Digital Engineering Faculty
Universität Potsdam
Information Systems Group

December 5, 2017

A thesis submitted for the degree of
"Doctor Rerum Naturalium"
(Dr. rer. nat.)
in Computer Sciences

Reviewers

Prof. Dr. Felix Naumann

Universität Potsdam, Potsdam

Prof. Dr. Ralf Schenkel

Universität Trier, Trier

Prof. Dr.-Ing. Ernesto William De Luca

GEI - Leibniz-Institute for international Textbook Research, Braunschweig

Published online at the
Institutional Repository of the University of Potsdam:
URN urn:nbn:de:kobv:517-opus4-412576
http://nbn-resolving.de/urn:nbn:de:kobv:517-opus4-412576

Abstract

With recent advances in the area of information extraction, automatically extracting structured information from a vast amount of unstructured textual data becomes an important task, which is infeasible for humans to capture all information manually. Named entities (e.g., persons, organizations, and locations), which are crucial components in texts, are usually the subjects of structured information from textual documents. Therefore, the task of named entity mining receives much attention. It consists of three major subtasks, which are named entity recognition, named entity linking, and relation extraction.

These three tasks build up an entire pipeline of a named entity mining system, where each of them has its challenges and can be employed for further applications. As a fundamental task in the natural language processing domain, studies on named entity recognition have a long history, and many existing approaches produce reliable results. The task is aiming to extract mentions of named entities in text and identify their types. Named entity linking recently received much attention with the development of knowledge bases that contain rich information about entities. The goal is to disambiguate mentions of named entities and to link them to the corresponding entries in a knowledge base. Relation extraction, as the final step of named entity mining, is a highly challenging task, which is to extract semantic relations between named entities, e.g., the ownership relation between two companies.

In this thesis, we review the state-of-the-art of named entity mining domain in detail, including valuable features, techniques, evaluation methodologies, and so on. Furthermore, we present two of our approaches that focus on the named entity linking and relation extraction tasks separately.

To solve the named entity linking task, we propose the entity linking technique, BEL, which operates on a textual range of relevant terms and aggregates decisions from an ensemble of simple classifiers. Each of the classifiers operates on a randomly sampled subset of the above range. In extensive experiments on hand-labeled and benchmark datasets, our approach outperformed state-of-the-art entity linking techniques, both in terms of quality and efficiency.

For the task of relation extraction, we focus on extracting a specific group of difficult

relation types, business relations between companies. These relations can be used to gain valuable insight into the interactions between companies and perform complex analytics, such as predicting risk or valuating companies. Our semi-supervised strategy can extract business relations between companies based on only a few user-provided seed company pairs. By doing so, we also provide a solution for the problem of determining the direction of asymmetric relations, such as the ownership_of relation. We improve the reliability of the extraction process by using a holistic pattern identification method, which classifies the generated extraction patterns. Our experiments show that we can accurately and reliably extract new entity pairs occurring in the target relation by using as few as five labeled seed pairs.

Zusammenfassung

Mit den jüngsten Fortschritten in den Gebieten der Informationsextraktion wird die automatisierte Extrahierung strukturierter Informationen aus einer unüberschaubaren Menge unstrukturierter Textdaten eine wichtige Aufgabe, deren manuelle Ausführung unzumutbar ist. Benannte Entitäten, (z.B. Personen, Organisationen oder Orte), essentielle Bestandteile in Texten, sind normalerweise der Gegenstand strukturierter Informationen aus Textdokumenten. Daher erhält die Aufgabe der Gewinnung benannter Entitäten viel Aufmerksamkeit. Sie besteht aus drei groen Unteraufgaben, nämlich Erkennung benannter Entitäten, Verbindung benannter Entitäten und Extraktion von Beziehungen .

Diese drei Aufgaben zusammen sind der Grundprozess eines Systems zur Gewinnung benannter Entitäten, wobei jede ihre eigene Herausforderung hat und für weitere Anwendungen eingesetzt werden kann. Als ein fundamentaler Aspekt in der Verarbeitung natürlicher Sprache haben Studien zur Erkennung benannter Entitäten eine lange Geschichte, und viele bestehenden Ansätze erbringen verlässliche Ergebnisse.Die Aufgabe zielt darauf ab, Nennungen benannter Entitäten zu extrahieren und ihre Typen zu bestimmen. Verbindung benannter Entitäten hat in letzter Zeit durch die Entwicklung von Wissensdatenbanken, welche reiche Informationen über Entitäten enthalten, viel Aufmerksamkeit erhalten. Das Ziel ist es, Nennungen benannter Entitäten zu unterscheiden und diese mit dazugehörigen Einträgen in einer Wissensdatenbank zu verknüpfen. Der letzte Schritt der Gewinnung benannter Entitäten, die Extraktion von Beziehungen, ist eine stark anspruchsvolle Aufgabe, nämlich die Extraktion semantischer Beziehungen zwischen Entitäten, z.B. die Eigentümerschaft zwischen zwei Firmen.

In dieser Doktorarbeit arbeiten wir den aktuellen Stand der Wissenschaft in den Domäne der Gewinnung benannter Entitäten auf, unter anderem wertvolle Eigenschaften und Evaluationsmethoden. Darüberhinaus präsentieren wir zwei Ansätze von uns, die jeweils ihren Fokus auf die Verbindung benannter Entitäten sowie der Aufgaben der Extraktion von Beziehungen legen.

Um die Aufgabe der Verbindung benannter Entitäten zu lösen schlagen wir hier die Verbindungstechnik BEL vor, welche auf einer textuellen Bandbreite relevanter Begriffe agiert und Entscheidungen einer Kombination von einfacher Klassifizierer aggregiert.

Jeder dieser Klassifizierer arbeitet auf einer zufällig ausgewählten Teilmenge der obigen Bandbreite. In umfangreichen Experimenten mit handannotierten sowie Vergleichsdatensätzen hat unser Ansatz andere Lösungen zur Verbindung benannter Entitäten, die auf dem Stand der aktuellen Technik beruhen, sowie in Bezug auf Qualität als auch Effizienz geschlagen.

Für die Aufgabe der Extraktion von Beziehungen fokussieren wir uns auf eine bestimmte Gruppe schwieriger Beziehungstypen, nämlich die Geschäftsbeziehungen zwischen Firmen. Diese Beziehungen können benutzt werden, um wertvolle Erkenntnisse in das Zusammenspiel von Firmen zu gelangen und komplexe Analysen ausführen, beispielsweise die Risikovorhersage oder Bewertung von Firmen. Unsere teilbeaufsichtigte Strategie kann Geschäftsbeziehungen zwischen Firmen anhand nur weniger nutzergegebener Startwerte von Firmenpaaren extrahieren. Dadurch bieten wir auch eine Lösung für das Problem der Richtungserkennung asymmetrischer Beziehungen, beispielsweise der Eigentumsbeziehung. Wir verbessern die Verlässlichkeit des Extraktionsprozesses, indem wir holistische Musteridentifikationsmethoden verwenden, welche die erstellten Extraktionsmuster klassifizieren. Unsere Experimente zeigen, dass wir neue Entitätenpaare akkurat und verlässlich in der Zielbeziehung mit bereits fünf bezeichneten Startpaaren extrahieren können.

Acknowledgements

This work would not have been done without the support of many people inside and outside of Hasso-Plattner-Institut. I would like to thank my advisor, Prof. Felix Naumann, for the patient guidance and useful advice. I have been so lucky to have a supervisor who taught me how to do research and gave me the chance of working on industry projects. Many thanks to my mentor, Dr. Gjergji Kasneci, for bringing me into this interesting field and providing me useful guidance during my Ph.D. studies.

I want to thank all my colleagues at the chair, their feedback helped me immensely towards this thesis. It was my pleasure to work in such a great team. In particular, I would like to thank, Dr. Ralf Krestel, for his guidance and collaboration. I am also grateful for the cooperation with Toni Grütze and Micheal Loster. I had a nice time to work with them. Besides, I would like to particularly thank Maximilian Jenders for all his help.

Most importantly, I am deeply moved by the support of my family and friends. You gave me the strength to complete this thesis. I would like to thank my parents for supporting me spiritually throughout my years of study and through the process of researching and writing this thesis. I would like to thank my wife, Yuxin Gao. This thesis is dedicated to her for all her love, encouragement, patience, and supporting.

Thank you very much, everyone!

Contents

CHAPTER 1

Named Entity Mining

In recent years, the data explosion has received much attention, and an enormous amount of digital data has been created. As the prediction result reported by Gantz and Reinsel (2012), from 2005 to 2020, the digital universe (i.e., the measure of all the digital data created, replicated, and consumed in one year) will grow from 130 exabytes to 40,000 exabytes. Nowadays, people are getting used to gaining information from the web, for example, tracking latest news, reading emails, and watching online videos. For commercial purposes, information contained in data is also important to improve and optimize decisions and performances. However, the useful information is usually hidden in data. To gain the targeted information from data, one needs to further process data. Considering the rapid increase in the volume of available data, it is getting infeasible to capture all targeted information manually. It brings the challenge that is to extract and maintain information from huge volumes of data automatically. Moreover, although no precise result of quantitative research has been published, most of data is unstructured, e.g., texts, audios, videos, according to some studies and observations. A common view is that an even larger proportion of data will be unstructured in the future. Due to the unstructured nature, it is difficult for humans, as well as machines, to gain insights from unstructured data, especially when the volume of data is extremely large. However, with the growth of unstructured data, the task of extracting structured information for further usages becomes more and more important.

From different viewpoints of various communities, the definition of structured and unstructured data can be different. For instance, HTML pages are considered as structured by Soderland (1997), but unstructured by Elmasri and Navathe (2003). Figure 1.1 represents an overview of the structurization of data based on the summarization by Chang et al. (2006). As the figure shows, the information stored in databases is considered to be structured, which is easiest to be maintained by machines. Free texts are the most difficult data for machines to understand, which require further processes and analytics to make the included information accessible.

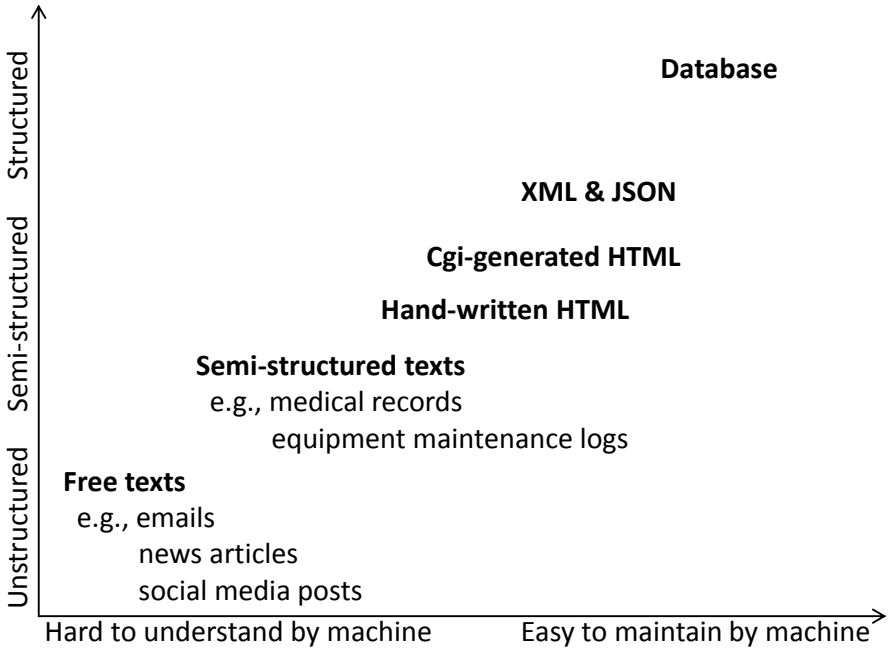


Figure 1.1: Structurization of various data, inspired from [Chang et al. (2006)]

Many researchers are focusing on the area of *Information Extraction (IE)*, which is the task of extracting structured information from unstructured or semi-structured data, including text documents and non-text documents. According to various inputs and the targeted outputs, many subtasks can be defined, such as named entity mining, terminology extraction, or template-based music extraction.

In this thesis, we focus on extracting information from free texts, more specifically, the task of named entity mining. Three major subtasks are involved, which are *Named Entity Recognition (NER)*, *Named Entity Linking (NEL)*, and *Relation Extraction (RE)*. These three tasks are all typical topics in the field of *Natural Language Processing (NLP)*. In the domain of information extraction, a named entity is a real-world object that can be denoted by a proper name. During the *Message Understanding Conference (MUC)*-6 [Grishman and Sundheim (1996)], the term named entity was defined, which contains persons, locations, organizations, and so on. The three subtasks can build up a complete pipeline to extract relevant structured information regarding mentions of named entities from texts. The NER task is aiming to recognize mentions of name entities and classify their into different types, such as person, location, organization. Then, NEL approaches disambiguate the recognized named entities and link them to the corresponding entities in a knowledge base. Finally, RE approaches extract relations between mentioned named entities based on contextual information.

The results of named entity mining, as well as each single subtask, can be used in many domains, such as natural language understanding, knowledge base population, and business intelligence. As an example use case, the volume of digital documents about business is growing rapidly. Every day, many more textual documents are generated by news agencies, companies, individuals, and so on. In these documents, rich information about business relations between companies is contained, for example, texts-fownership_of, competitor_of, and customer_of relations. This valuable information can be extracted via a named entity mining system. The business relations build a business network of companies. Suppose a bank try to estimate the risk of giving loans to enterprise customers. Except for some standard risk prediction models, the business network can give a better view of their customers. Instead of analyzing one single company, the bank can predict the risk more precisely by taking the conditions of other closely related companies into account. For instance, a company could run into trouble when the major customers of the company perform badly.

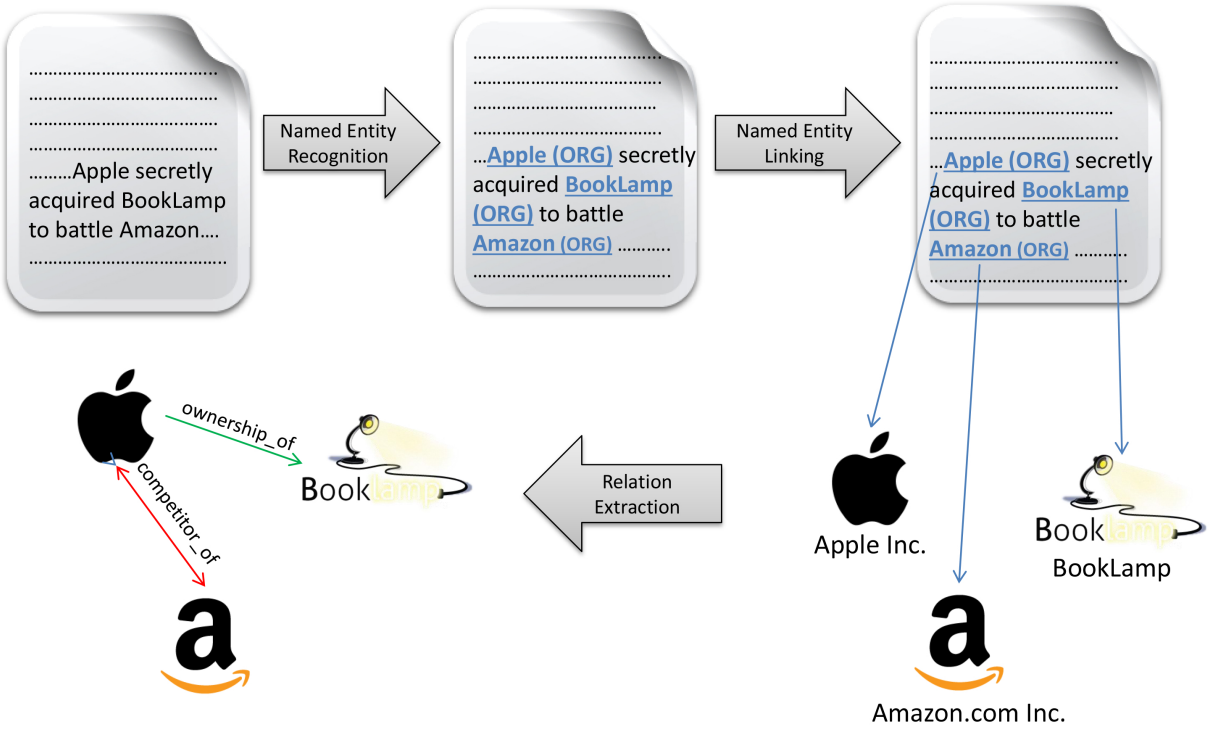


Figure 1.2: Example of named entity mining process

To explain an example of the named entity mining process, assume a text document that might contain the following sentence:

... Apple secretly acquired BookLamp to battle Amazon. ...

The first step is to apply named entity recognition for extracting mentions of named

entities and identifying their types. With this fundamental process, three organizations, Apple, BookLamp, and Amazon can be extracted for the example sentence. In the next step, named entity linking, we try to disambiguate and link each mention to the corresponding named entity in a *Knowledge Base (KB)*. Suppose we use a KB containing all referred organizations to this example, we can link the recognized mentions to the companies *Apple Inc.*, *BookLamp*, and *Amazon.com Inc.* Finally, we can identify the relations between these companies via relation extraction approaches. Based on the given context, we know that the company *Apple Inc.*, which is the parent company of *Booklamp*, is a competitor of *Amazon.com Inc.* Figure 1.2 presents the step by step processing of named entity mining on the above example. In this figure, the final output of such a system can naturally build a semantic network based on the relations among these three companies.

It is worth to note that context is the key to solving the named entity mining problem. Relevant information about named entities is given in the contexts of their mentions. Analyzing a mention by either humans or machines is impossible without any contextual information. In the previous example, if the term "Apple" is given alone, one can not identify what this term refers to. It could refer to the fruit, a person, a company, a location, etc. Therefore, most of the state-of-the-art approaches rely on the information delivered by contexts to analyze named entities. We introduce these three subtasks in the following sections in detail.

1.1 Named entity recognition

Named entity recognition is also known as entity identification, entity classification, and entity extraction. As mentioned by Nadeau and Sekine (2007), the NER task was first defined during MUC-6 [Grishman and Sundheim (1996)], where the objective was to discover general entity types, such as persons, locations, organizations as well as time, currency, and percentage expressions from texts. Except for these general types of named entities, it is also interesting to recognize other types of named entities according to further usages. For example, location can be further classified into subcategories, including city, country, street, and so on. In the domain of biology, the NER task is defined to recognize mentions of protein, DNA, RNA, etc. As a fundamental task of NLP, the result of NER can be used for many further applications, such as entity linking, text summarization, and question answering.

The NER task can be further separated into two phases, which are detection mentions of named entities from texts, and classification of mentions by the types of corresponding named entities. These two steps bring the challenges of NER problem.

First, one needs to find out mentions of named entities from texts. In particular languages, some significant features are strong indicators of mentions. For example, in English texts, the first letter of a mention is always capitalized. However, it is not

adaptable to other languages, e.g., German, since each noun starts with a capital letter in German.

Second, mentions often consist of multiple terms. In this case, an NER approach needs to extract the complete mention correctly, For example, if the mention "Bank of China", which refers to an organization, is wrongly recognized to be "China", an NER approach might identify it as a location by mistake.

Finally, the ambiguity of mentions also can lead to wrong NER results. By definition, an NER approach does not need to identify which specific named entity a mention refers to. However, classifying entities with the correct types is the core of NER. For example, the mention "Jordan" can refer to a country, a person, or even an organization. It is more difficult to classify the type of an ambiguous entity. In Chapter 2, we introduce the state-of-the-art of named entity recognition approaches.

1.2 Named entity linking

Named entity linking is the task of establishing a mapping from textual mentions of named entities to canonical representations of those entities in a knowledge base, such as DBpedia [Auer et al. (2007)], Freebase [Bollacker et al. (2008)], or YAGO [Hoffart et al. (2011a)]. Often, textual mentions are ambiguous; that is, a mention could refer to multiple named entities, but only one of them is the correct one in the given textual context. Resolving these ambiguities is often referred to as *Named Entity Disambiguation (NED)*, which is a highly challenging aspect of an NEL process. More specifically, a robust NEL algorithm has to robustly resolve ambiguities and thus build on robust NED methods. The NED problem is often ill-posed, as only the right context and background knowledge can help disambiguate entities. As an example, consider the sentence: "*London spent* $\$80,000$ *(*$\$2,040,000$ *in current value) to build a* $15,000$*-square-foot stone mansion ('Wolf House') on the property.*" A human reader knows that in general money is spent by people, but sometimes also city councils can spend money, and hence, in the above sentence "London" may refer to a person by that name or to the capital of Great Britain. When considering the contextual information, especially the key phrase "Wolf House", and the fact that this was the name of the mansion of the writer Jack London, the disambiguation of "London" becomes obvious. In many cases, however, the contextual information is implicit and may be latently spread across various passages or documents, and background knowledge may not be sufficient, which makes the disambiguation task challenging even for human readers. Many further applications are based on results of NEL approaches, such as semantic search, machine translation, business intelligence, topic detection, text summarization, machine vision, and many more.

It is worth to note that the NED problem is abundant. In the context of information systems, the problem has been addressed in many different flavors and settings, e.g.,

in the structured setting of record-linkage and duplicate detection, where the goal is find database records that refer to the same named entity [Bhattacharya and Getoor (2007); Naumann and Herschel (2010)], in the semi-structured setting of cleaning XML data [Weis and Naumann (2005)] or annotating Web tables [Limaye et al. (2010)], in the context of enriching Wikipedia information boxes [Wu and Weld (2008)], for the alignment of knowledge bases [Aumueller et al. (2005); Lacoste-Julien et al. (2012)], and most prominently, in the setting of Natural Language Processing [Bagga and Baldwin (1998); Bunescu and Mooney (2005a); Cucerzan (2007); Fleischman (2001); Mann and Yarowsky (2003)], which is also the setting of focus in this thesis.

State-of-the-art approaches commonly solve the NEL problem in three major steps, namely candidate entity selection, candidate entity ranking, and NIL entity prediction.

As a initial step, candidate entity extraction is to find out all probable entities that a mention can refer to in a selected KB. Generally speaking, a KB covers as much information as possible, so that it contains a large number of named entities. Most mentions can refer to only a few of them. Even in extremely ambiguous cases, most of the named entities in KB are irrelevant to such a mention. Therefore, it is important to shrink the search space in the early stage and keep only the reasonable candidates for the next disambiguation step.

With a candidate entity list, the next step is to link the mention to the most probable entity that the mention can refer to in the candidate list. This step is the core of NEL, which is the disambiguation process of mentions with the given contexts. It is often formulated as a ranking problem. According to some ranking strategies, entities in a candidate list of a mention are ranked. The most relevant candidate should be top-ranked and linked to the mention. For instance, "Jordan" is mentioned in sentence "*Jordan is an American retired professional basketball player, who has won five MVP awards.*" In this case, an NEL approach should rank the entity of the famous NBA basketball player, *Micheal Jordan*, in the top position, and link this mention to it.

Finally, although KBs typically contain lots of entities, there are still many named entities that are not covered. Therefore, it is also important to identify whether a referred named entity is included in the exploited the KB. When the named entity is not included, the mention is linked to the NIL entity. It is often designed as a post-process of candidate ranking. However, some strategies also combine the two steps by including the NIL entity in each of candidate lists and ranking it together with other candidates.

The major challenge of the NEL task is the disambiguation process. It is a difficult task, especially when a mention can refer to multiple named entities that are similar to each other. As an example, *George H. W. Bush* and *George W. Bush* were both the presidents of the United States. To automatically identify whom the mention "George Bush" refers to is extremely hard since they have too much in common. Moreover, information contained in KBs does not always be able to match the one delivered by contexts or sometimes key information for disambiguation can be missing in both sources.

In Chapter 3, we introduce our solution for named entity linking task.

1.3 Relation extraction

Relation extraction is the task of detecting and classifying semantic relations between named entities from texts. In the setting of NLP, relations denote the semantic connections between named entities, which are interesting structured information contained in texts. RE is crucial for natural language understanding and plays a major role in transforming unstructured data to structured information. Most RE systems focus on extracting binary relations, which means relations between two named entities. The problem of n-ary relations extraction, where n is greater than 2, receives much attention with the development of RE. It is sthtraightforward to build up a complex semantic network of named entities with the information of relations as presented in Figure 1.2. Based on such a network, one can have a high-level overview of relational information contained in the given textual sources. Furthermore, more knowledge about relevant named entities can be derived from extracted relations. For example, an RE approach extracts two relations: Person A is the father of Person B, and Person C is the son of Person B. Although the relation between Person A and Person C is not given, it is easy to reason that Person A is the grandfather of Person C.

Due the complex nature of texts, RE is a nontrivial task. RE approaches need to be able to capture the complex semantic information from contexts of mentions.

The first challenge is the diversity of relations types. Each possible combination of two named entity types can have various relation types. In MUC-7 [Chinchor and Marsh (1998)], as a subtask of the information extraction, three target relation types, i.e., employee_of, location_of, and product_of are introduced. In the *Automatic Content Extraction (ACE)* program, more relation types are taken into consideration. The ACE2003 dataset [Strassel et al. (2003)] includes 24 relation types, e.g., part_of, parent, and located. Moreover, in specific domains, more relations types are defined according to different requirements.

Secondly, a pair of named entities can participate in various relations. A person can be born, live, visit, or die in a location. An organization can be a competitor, partner, or subsidiary of another one. It is difficult to classify in which type of relation the entity pair participates according to the context. Even worse, an entity pair can participate in multiple relations at the same time. A straightforward example is that a person can play various roles in an organization, such as co-founder and CEO.

Thirdly, some types of relations between entity pairs can change over time. For instance, when two persons are married, they have the relation wife_of or husband_of. But if they are divorced, the relation between them become ex-wife_of or ex-husband_of.

Except for the challenges caused by the complex nature of relations, some other factors

make the task of correctly extracting all relations between mentioned named entities from texts difficult. For a single relation, there are many different ways to describe it in context. As a simple example, all of the snippets "... *Google owns Youtube* ...", "... *Youtube, owned by Google* ...", and "... *Youtube is a subsidiary of Google* ...", describe the same fact that *Google* owns *Youtube*. Furthermore, the task is usually based on the result of NER and NEL processes, which means, the quality of the RE result is highly influenced by the named entity annotations in input documents. However, no system can provide 100% correct named entity annotations automatically, which will also mislead RE approaches. In Chapter 3, we present our RE approach to extract business relations between companies.

1.4 Structure and contributions

Figure 1.3 presents the architecture of a general named entity mining system based on the three introduced tasks. By giving a collection of text documents, the first step is to apply NER for recognizing mentions of named entities in texts. Based on the recognition result, the next step is to disambiguate the recognized mentions according to the corresponding contexts and link them to the entities in a KB. Afterwards, one can further extract the relations between entities that are mentioned in the given corpus. Please note that for the task of relation extraction, according to the further requirements, the disambiguation step (i.e., NEL) is optional. Some RE approaches directly extract relations between entities based on the results of NER.

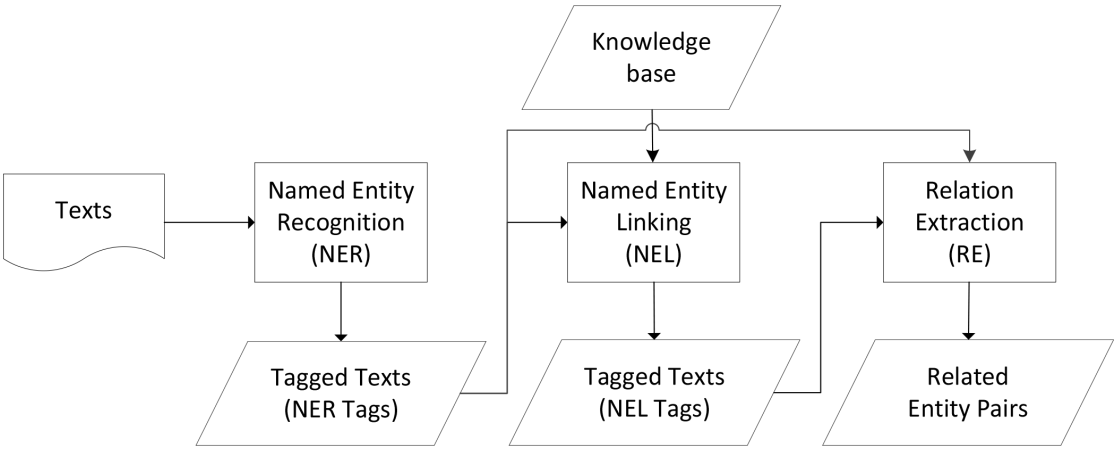


Figure 1.3: The architecture of a general named entity mining system

Considering the long research history of the topic of NER, many well-developed NER systems are available, which can produce reliable named entity annotations, e.g., Stanford NER [Finkel et al. (2005)], and Apache OpenNLP[1]. Therefore, in this thesis, we

[1] https://opennlp.apache.org/

mainly focus on the other two tasks, NEL and RE. We aim to provide light-weighted solutions of these problems, which can be applied on web-scale corpus and produce reliable results efficiently.

In summary, the structure and the main contributions of this thesis are:

Chapter 2 is a survey of named entity mining research, where we separately present a comprehensive review of various aspects of each subtask, including NER, NEL, and RE. For each task, we discuss the development of the domain, summarize the frequently exploited features for solving the problem, and introduce the-state-of-the-art techniques. We also introduce the general evaluation methodologies and important benchmark datasets in this chapter.

In Chapter 3, we introduce our named entity linking approach BEL [Zuo et al. (2014)]. BEL includes an ensemble-based disambiguation approach that exploits the terms that surround a textual mention to best capture its context, and a parsimonious linking model that combines the above method with a prior probability of a candidate named entity being referred to by a given mention yields a highly efficient linking process. Furthermore, BEL exploits bagging strategy to improve the performance of the approach. The experimental results show BEL is an efficient NEL approach that can provide reliable entity linking results. We also introduce a version of BEL, which exploits a dictionary contains only company entities as the knowledge base. In this version, we simplified BEL to further improved the efficiency to disambiguate company mentions in large amount of textual data. This version is used as the preprocessing of our business relation extraction approach, which is introduced in the next chapter.

In Chapter 4, we introduce our relation extraction approach, which focuses on extracting business relations between companies [Zuo et al. (2017)]. To solve this problem, we present a novel semi-supervised relation extraction approach, which requires only a small amount of manually specified company pairs to extract new ones that belong to the targeted semantic relation. We also provide a straightforward solution to identify the direction of asymmetric relations reliably. We further define a holistic pattern identification strategy, which enables us to extract multiple relation types simultaneously.

Finally, we summarize the results of this thesis and discuss the future work in the domain of named entity mining in Chapter 5.

CHAPTER 2

State-of-the-art for Named Entity Mining

As we have already mentioned in the previous chapter, named entity mining is one of the most interested topics of information extraction. The focus of this task is on extracting valuable information about named entities mentioned in texts. A named entity plays an important role in extracting structured information from textual documents. With the three tasks, *Named Entity Recognition (NER)*, *Named Entity Linking (NEL)*, and *Relation Extraction (RE)*, we can automatically extract mentions of named entities from texts and identify the types of named entities, the specific named entities to which mentions refer, and relations between named entities.

In this chapter, we introduce the relevant aspects of NER, NEL, and RE, as well as the state-of-the-art techniques in detail. Since all three tasks are aimed at extracting information about named entities, they are highly correlated to one another. Therefore, we also summarize the common techniques that are applied for solving these problems.

The structure of this chapter is organized as follows: We introduce the techniques and evaluation methodologies for these tasks in Section 2.1. In Sections 2.2, 2.3 and 2.4, we introduce NER, NEL, and RE one by one in detail. Finally, Section 2.5 represents the entire pipeline of a named entity mining system based on our NEL and RE approaches.

2.1 Common techniques

Machine learning techniques are widely applied for solving NER, NEL, and RE problems. In this thesis, based on the requirement of labeled input data, we separate the techniques into three categories in general: *supervised, unsupervised,* and *semi-supervised learning.*

Supervised learning: In a supervised learning scenario, a training set is required as the input of a system. A training set is often obtained manually by human experts according to the targeted task. Based on the training set, a supervised learning algorithm tries to infer a function that can be applied to new unlabeled examples.

Many supervised learning approaches are applied for solving *Natural Language Processing (NLP)* tasks, for example *Decision Trees (DT)*, *Hidden Markov Model (HMM)* [Rabiner (1989)], *Support Vector Machine (SVM)* [Cortes and Vapnik (1995)], *Maximum Entropy (MaxEnt)* [Berger et al. (1996)], *Conditional Random Fields (CRF)* [Lafferty et al. (2001)], and *Convolutional Neural Networks (CNNs)* [Schmidhuber (2015)]. These supervised learning techniques have been able to produce good results. However, one obvious limitation, which cannot be avoided, is the requirement of labeled training data. The labeling process is often time-consuming, and sometimes knowledge of domain experts is required. In this process, the regulation of labeling needs to be predefined according to the aimed task. Therefore, a labeled training set typically can be used for only one specific task. When the task is changed, or a new task is defined, the training set needs to be relabeled to adapt to the change. Furthermore, in some cases, the trained model can only be applied to the documents that are similar to the ones contained in the training set. For example, if a model is trained based on a biology-related text corpus, when it is used for texts from other domains, its performances could significantly decrease. Also, over-fitting can occur during the training process.

Unsupervised learning: The goal of unsupervised learning is to find structure in a given dataset, e.g., clustering the data points in a dataset. As expressed in the name, unsupervised learning strategies do not require any extra supervision from humans. An unsupervised learning-based approach does not require further information except the data, which need to be processed. The advantages of unsupervised learning strategies are to avoid a tedious hand-labeling process and that they do not need any predefined information. It is typically difficult to understand and evaluate the outputs of such a system. Especially in our case, the semantic meaning of each output cluster is unknown; therefore, further analytics on the outputs are required.

Semi-supervised learning: For many natural language processing tasks, bootstrapping is a commonly applied semi-supervised learning technique. Instead of providing large training sets or no labeled data at all, such as in the previous two strategies, a semi-supervised system starts with a small set of labeled data, which is called a seed set. According to the problem that a system needs to solve, users can give the system some examples as seeds. Comparing to labeled data for training, it is usually much simpler to label several examples. Then the system inspects all textual documents contained in a given corpus and extracts occurrences of the seeds. Common contextual clues are used to find more instances that appear in similar contexts from the entire corpus. Thereafter, the system repeats the same process, including the newly extracted instances in the seed set. With iterations, more and more instances can be gathered until the system is terminated according to some predefined rules. One of the major disadvantages of semi-supervised learning is that, when wrong examples are included during iterations,

serious semantic drift problems can happen, which makes the entire process fail.

2.2 Named entity recognition

In this section, we introduce various aspects of the named entity recognition task. First, we present the types of named entity. Second, we introduce some useful features and techniques for solving the NER task. Third, we discuss state-of-the-art approaches that focus on recognizing named entities from social media data. Finally, some commonly used benchmark datasets are introduced.

2.2.1 Named entity types

As introduced in Chapter 1, MUC-6 [Grishman and Sundheim (1996)] firstly defined the general entity types, such as named entity expressions (ENAMEX) (i.e., person, location, and organization), time expressions (TIMEX) (i.e., time and date), and numeric expressions (NUMEX) (i.e., percentage and money). Those general types of name entities are most frequently mentioned in textual documents. Therefore, lots of approaches focus on identifying the mentions of named entities with these general types, especially the three major ones, person, location, and organization.

Named entities can be classified into not only the general types but also other ones, e.g., fine-grained types. According to the targeted information for further usages, it is also important to be able to identify named entities with predefined types.

One straightforward thought is that of extending the basic types of named entities by focusing on the fine-grained subcategories. Fleischman (2001) presented a partially automated system to classify location instances into eight subcategories: city, country, street, region, water, artifacts, territory, and mount. Compared with locations, the person class is more difficult to split into subcategories, since it has no obvious common rules. One possible method of defining this class is according to its further usages. Subsequently, Fleischman and Hovy (2002) tried to classify mentions of persons also into eight subcategories, which are athlete, politician/government, clergy, businessperson, entertainer/artist, lawyer, doctor/scientist, and police. These subcategories are defined based on their occurrence frequency and usefulness in the potential application, i.e., a question-answering system. Chinchor (1998) further defined organizations to be government, company, and others. The task of recognizing fine-grained types of named entities is very challenging for both machines and humans, since there is often a lack of detailed information for identification in contexts. One example sentence that is mentioned by Fleischman and Hovy (2002) is as follows: *"It's dangerous to be right when government is wrong", Lrsyomh told reporters.* One can easily figure out that Lrsyomh refers to a person but hardly identify that Lrsyomh is a businessman since no clear clue is given in the context. Even worse, the surrounding terms included in this given sentence

can mislead an NER approach to get the wrong result that Lrsyomh is a politician.

In some specific domains, especially biology, the important entities in relevant documents are not limited to persons, locations, and organizations anymore. Some special types of named entities have to be recognized. In the biology domain, the NER task focuses on recognizing biologically meaningful terms. Kim et al. (2003) introduced the GENIA corpus, which consists of nearly 100,000 annotations in 36 different classes of biological terms, including entity types such as protein, DNA, and RNA. Those terms are defined hierarchically. For example, the names of proteins are further classified into seven subcategories such as molecule, complex, and family of group. Based on GENIA, Kim et al. (2004) introduced the JNLPBA shared task of bio-entity recognition. They simplified the original annotations in GENIA into five classes: protein, DNA, RNA, cell line, and cell type. Moreover, in the biomedical domain, recognition of disease or medicine names from texts is as important task [Jimeno et al. (2008); Leaman and Gonzalez (2008)].

Furthermore, according to contents of the given documents and specific requirements, some NER approaches are also developed to recognize customized types of named entities. Maynard et al. (2001) introduced MUSE for processing texts from widely differing domains and genres. MUSE can process multiple types of texts in a robust manner. In this work, they also extended the general named entity type by including address entities, including email, URL, telephone, and IP address. Etzioni et al. (2005) introduced the NER process of their IE system KNOWITALL, which can also recognize film titles. By classifying this specific type, KNOWITALL can extract the relations between actors and films in the following steps.

2.2.2 Features for NER

Extraction of important features is crucial in solving the problem of recognizing named entities from texts. In this section, we mainly focus on the frequently employed features of the NER problem from English texts in three different categories, namely *local features*, *global features*, and *external features*, which are used by both rule-based and machine-learning-based approaches.

Local features

Local features refer to the ones that are derived directly from terms in surrounding contexts. For each mention of a named entity, these features can be extracted from terms of the mention or the neighboring terms (e.g., the previous term and the following one).

Character-level features: A group of features is based on characters in terms. This group is one of the most straightforward ones for the NER task. Table 2.1 lists several example features that were introduced by Bikel et al. (1997) as well as Chieu and Ng (2002). For instance, whether characters in a word are capitalized is typically a significant indicator of mentions of named entities: When a term starts with a capital letter, the probability of the term being a mention of a named entity is high, and a term that contains or ends with a period can be the short form of a proper name. In the case that terms contain digits, one can also easily predefine some patterns for recognition. For example, a number that is followed by a percent symbol stands for a percentage. McCallum and Li (2003) also defined 16 regular expression patterns of extracting mentions of named entities, i.e. A, $A+$, $Aa+$, $Aa+Aa*$, $A.$, and $D+$, where A, a, D respectively indicate capital letters, small letters, and digits.

Table 2.1: Example of character-level features

Feature	Example Word	Explanation
AllCap	SAP	Contains only capital letters
InitCap	Einstein	Starts with a capital letter
MixCap	McKesson	Contains mixed case letters
Period	I.B.M.	Contains periods
LetterAndDigit	W3C	Contains letters and digits
Digit	2017	Contains only digits
DigitAndSlash	2017/01/01	Contains symbols and digits
DigitAndPercent	25%	Contains symbols and digits

Position in a sentence: The information of a term's position in a text can also be used as a feature. For example, whether a term is at the beginning of a sentence is a useful feature. Together with the character-level features (e.g., capitalization of the first character), the position can avoid wrong annotations caused by the fact that all terms at the beginning of an English sentence start with a capital letter.

Part-of-speech (POS) tagging: POS tagging is the process of labeling terms in a text with a particular POS. The commonly used POS tags in English are noun, verb, article, adjective, preposition, pronoun, adverb, conjunction, and interjection. The POS tag of each term are widely used in many approaches as essential features, e.g., in Stanford NER [Finkel et al. (2005)].

Prefixes and suffixes: These are two strong indicators of named entities, which can be used as features. Chieu and Ng (2002) exploited two features, *Corporate-Suffix* and *Person-Prefix* in their approach based on this information. These features were collected from the training data that were used in these researchers' work. Each of the features

consists of a list of tokens. The list for *Corporate-Suffix* contains tokens such as "ltd.", "associate", and "corp." If an n-gram ends with the tokens in this list, it has a high probability of being a corporation's name. Similarly, if a term occurs in a piece of text after a token such as "Mr.", "Dr.", and "chairman", it probably refers to a person.

The features that are introduced above can provide useful information for recognizing named entities; many NER approaches are designed based on these features, especially the rule-based ones, in the early stage. However, one significant limitation is the language dependency: Some of the features can be applied to only one or more specific languages. For example, the capitalization of the first character works well for English texts but not for texts in other Roman languages. In German texts, all nouns begin with a capital letter. As another example, even when we focus on English texts, in British English, the date is typically expressed in the order "day-month-year", but the order in American English is "month-day-year". Thus, the pattern design for British English can mistakenly recognize "01-02-2017" to be January 2^{nd}.

Global features

The information contained in an entire document is valuable for the NER task, and some useful features can be extracted. Compared to the local features, which only focus on a mention and its surrounding context, global features capture information contained in a document.

Multiple occurrences: A named entity, especially the most important one for a document, is often mentioned multiple times in the document. By summarizing the information of all occurrences of the same named entity, an NER system can make a joint decision and avoid individual mistakes. For instance, suppose one entity is mentioned three times in one document, and an NER system recognizes two mentions referring to a person, but the other mention is mistakenly recognized as an organization since the local context contains information that misleads the system. In such a case, with the global information, the system can correct the incorrect label according to annotations of the other two occurrences.

Another approach was introduced by Cucerzan (2007). As we have mentioned, all terms at the beginning of an English sentence should start with a capital letter. In this case, we cannot directly identify whether those terms are mentions of named entities according to the capitalization information. Cucerzan (2007) introduced a strategy that uses global information. When a mention occurs in both the uppercased and the lowercased forms in a single document, it is considered to be a normal noun other than a proper name.

In the simple case that a named entity is mentioned in the same form in a single document, one can easily match the mentions. In more complex cases, the mentions

of a unique entity can be different. However, by defining the rules properly, one can still make use of the information for the NER task. For instance, Chieu and Ng (2002) introduced a feature called acronyms. They regarded the terms that were made up of only capital letters as acronyms. Then, their approach tried to find the sequence of terms whose starting characters can match the acronym. For example, if "IBM" and "International Business Machines" are two mentions in one document, both of them are labeled as mentions that refer to the same named entity. The approach of Mikheev et al. (1998) annotates new names when they contain an ordered subset of another name that has already been recognized as a named entity. As an original example from the paper, if "Hughes Communication Ltd" has already been recognized as an organization, the mention "Hughes" should also refer to the same organization.

Position in a document: A document can consist of different parts, such as a headline, an author field, and the main body. The position at which a named entity is mentioned can also be used as a feature. Minkov et al. (2005) presented an approach to extracting person names from emails. They introduced special features for emails, and some of these features are based on the position information. In their work, all email documents are assumed to have specialized structural fields, including *From*, *Subject*, *Time*, *Signoff*, and so on. In this specific example, the named entities contained in the fields of *From* and *Signoff* are typically of the type person name.

Meta-information: Nadeau and Sekine (2007) introduced the use of meta-information of documents as features. For instance, if we know the given document is an email, the probability of extracting person names as well as email addresses is higher compared to that of extracting them from normal texts. Similarly, in news articles, the author name and locations are expected to be mentioned; in financial reports, company names are frequently mentioned.

Global features are useful for NER, but the limitations are also evident. Most of them are valid under some specific conditions, e.g., multiple mentions of the same named entities, and extra information about the type of the given documents. Furthermore, using global information can sometimes cause errors. For instance, when an ambiguous mention that is contained in one document multiple times refers to different types of named entities, with global features, an NER approach can wrongly assign one unique type to all occurrences.

External features

Gazetteers are one of the most frequently exploited external resources. It consists of a set of lists, which can contain person names, locations, organizations, etc. Mikheev et al. (1999) investigated the importance of gazetteers. Their gazetteer includes around 4,900 location names, 30,000 organization names, and 10,000 first names of persons.

As the experimental result shows, the performance significantly improved because they included this gazetteer, especially for recognizing location names. In fact, it is simpler to build up and maintain a list of location names compared to other types of lists, since location names are relatively stable.

However, there is not always a proper gazetteer available for different NER tasks. Except for some entities, e.g., months, days of the week, and common abbreviations, specific gazetteers need to be generated. Some studies focused on generating gazetteers automatically. Hearst (1992) applied lexico-syntactic patterns to find nouns that belong to the same semantic class. For example, with a proper pattern, one can include France, England, and Spain in the list of European countries from the snippet "*... most European countries, especially France, England, and Spain ...*"

Riloff et al. (1999) applied the mutual bootstrapping technique to learn dictionaries from web pages starting with a small set of lexical patterns and entities. Nadeau et al. (2006) introduced an unsupervised strategy to generate gazetteers and employ them for NER. With their web-page wrapper approach, Nadeau et al. (2006) built up their gazetteer including 14,977 city names, 20,498 company names, 35,102 first names of people.

2.2.3 Recognition techniques

As a fundamental research topic in NLP, there is a long history of the development of NER. Rule-based approaches are often applied as a simple solution for NER. Learning strategies are also effective ways of solving the problem.

Rule-based method

Rule-based methods have frequently been applied in the NER task. Rules (e.g., grammar-based rules) can be used solely or together with other techniques for NER. In MUC-7 [Chinchor and Robinson (1997)], several top-ranked approaches according to their performance in the NER challenge exploit rule-based methods. For instance, Fukumoto et al. (1998) introduced the Oki system for MUC-7, which includes a pure rule-based NER system. It employs several predefined rules, e.g., extracting capitalized words as candidate named entities except for the ones that occur at the beginning of sentences. They also used lexico-syntactic patterns for classification of named entities. Black et al. (1998) presented an NER system that employs a certainty theory formula to combine certainties of multiple rules for the same entity. Mikheev et al. (1999) introduced an approach that combines rule-based grammars with statistical models to recognize named entities. The system starts by applying sure-fire grammar rules. An example rule pattern for recognizing organizations is "*Shares in Xxxx+*", where X indicates a capital letter and x represent a small one. With this rule, when the piece of text "*... shares*

in Apple ..." is contained in the given corpus, mention "Apple" is annotated to be an organization. The rule-based method has also been successfully applied in the biology domain [Hanisch et al. (2005)].

Discussion: Using rules to recognize a named entity is a straightforward solution. Most of the rules are derived from the intuitions or observations of how named entities can be mentioned in text. Many rule-based systems perform well on the NER task. However, the quality of NER results highly depends on the selection of rules. All instances that do not match the given rules are omitted; this can cause bad recall performance.

Supervised learning

Supervised learning techniques are frequently applied in the field of NER, which has been proved to deliver good results.

HMMs was first applied for NER by Bikel et al. (1997). Following the basic idea of HMM, the NER problem is formulated as follows: Given a sequence of words W, find a sequence of name-classes NC that maximizes the probability $P(NC|W)$. Bikel et al. (1999) further expanded the experiments. Zhou and Su (2002) introduced an approach based on a chunk tagger that maintains the extracted features. Except for commonly used features, for each word sequence, they included an NE-chunk (based on text chunking [Tjong Kim Sang and Buchholz (2000)]) that consists three parts: boundary category, entity category, and word feature. With the NE-chunk, one can build up some constraints between neighbor terms, for example, if the current term is at the beginning of an entity, the next term must be in the middle or at the end of an entity with the same entity category (i.e., entity type).

Borthwick (1999) introduced the MENE system in his Ph.D. research to recognize named entities using the maximum entropy approach with mainly local features. The MENE-Proteus system is also presented in the study, which employs another MaxEnt classifier using global information to correct mistakes. Afterwards, Chieu and Ng (2002) introduced the MENERGI approach, which can combine the typical local and global features in a single MaxEnt classifier. With the set of features, MENERGI tries to find a sequence of word classes that maximize $P(c_1, ..., c_n|s, D)$, where $c_1, ..., c_n$ are the classes of the words of a sentence s in a document D. The MaxEnt approach has also been applied in the biomedical domain. Lin et al. (2004) applied MaxEnt to recognize biomedical named entities. A morphological feature is included by MaxEnt, due to the specialty of the biomedical domain, since named entities in this domain typically contain some special prefixes, postfixes or suffixes, such as "anti-", "hemo-", and "vitamin".

In 2003, McCallum and Li (2003) explored CRF for the NER problem and published the preliminary results. The experimental results on the CoNLL-2003 dataset [Tjong Kim Sang and De Meulder (2003)] indicated that CRF performs well in the field of

NER. Settles (2004) empirically presented the performance of CRF to recognize protein, DNA, RNA, cell-line, and cell-type. The Stanford NER [Finkel et al. (2005)] is also a CRF-based system that is one of the most popular NER tools. Loster et al. (2017) focused on extracting company names from a German text using dictionaries of companies. Their experimental results revealed the influence of employing gazetteers together with the CRF method for the NER problem on this specific entity type.

Paliouras et al. (2000) used a decision tree to solve the NER problem. A benefit of applying DT is that manually generated grammars rules are not required. Without any such rules, which are manually generated following intuitions or experts experiences, the system can automatically learn rules according to the given training data. Paliouras et al. (2000) represented texts by feature vectors, in which each vector consists of 14 words from the context. Each of these words is represented by its POS and gazetteer tag.

Isozaki and Kazawa (2002) exploited Support Vector Machines to classify named entities into different classes. They focused on optimizing their SVM-based system (e.g., reducing the feature space) to improve the performance of efficiency.

Instead of using one single learning technique, multiple techniques can be employed at the same time on the same corpus and make the decision together in the end. Florian et al. (2003) combined four classifiers, namely the robust linear classifier [Zhang and Johnson (2003)], the MaxEnt classifier, the transformation-based learning classifier, and the HMM classifier [Bikel et al. (1999)], with different strategies (e.g., majority voting). The experimental result indicate that the performance of the combined system is significantly improved compared to that of the best performing single classifier.

Collobert and Weston (2008) introduced a *convolutional neural network (CNN)* architecture that can extract multiple NLP results, including POS tags, chunks, NER tags, etc. For the NER task, their approach can recognize a person, a company, and a location. Chiu and Nichols (2015) used *long short-term memory networks (LSTMs)* and CNN architecture to automatically generate word- and character-level features to solve the NER problem. Lample et al. (2016) represented their NER approach also based on LSTMs. By avoiding language-specific resources or features, their approach can produce good NER results in various languages.

Discussion: As we have introduced, many different supervised-learned techniques have been applied for recognizing named entities. These techniques are typical solutions of the NER problem. It is relatively easy to create labeled training data for NER, comparing to the other two tasks, named entity linking and relation extraction, since only mentions of named entities and their types need to be labeled. However, the requirement of training data is still one of the most significant limitations for supervised-learning approaches. When the targeted group of entity types changed, specific training set needs to be created.

Semi-supervised learning

Collins and Singer (1999) introduced the CoBoost to classify the types of named entities. Although they claimed that their approach was unsupervised, we consider it to be a semi-supervised approach, because CoBoost is based on a bootstrapping strategy that requires a hand-crafted seed set as the input. The seed set consists of spelling rules (i.e., example names). In their experiment, the initial words of spelling rules include "California", which is a location, "Microsoft", which is an organization, all mentions that contain "Incorporated" are organizations; and so on. With these predefined spelling rules, CoBoost first extracts contextual rules from the text to build up a set of contextual rules. Afterwards, by using these contextual rules, new spelling rules can be found and used in further iterations.

Instead of applying a single classifier, Carreras et al. (2003) exploited multiple semi-supervised classifiers for NER. With the AdaBoost algorithm, the decision of each classifier is combined into a weighted sum, which is the final output of the system.

Many semi-supervised approaches apply bootstrapping with some rule-based strategies. However, the bootstrapping strategy can also work together with supervised learning strategies. In a study by Liao and Veeramachaneni (2009), the NER system trains CRF models iteratively. Following the idea of bootstrapping, they train the first CRF model with a small set of labeled data, which are automatically generated using high-precision decision rules. With the trained model, new named entities can be recognized and then used to train new CRF models in following iterations. As the experiment indicates, the result of the researchers system that starts with 60 training documents outperforms the pure supervised learning approach trained with 60 or 700 documents.

Discussion: Please note that in this thesis, we distinguish semi-supervised and unsupervised learning according to whether any labeled data are provided as initial input data. Therefore, although some related works of NER (e.g., Collins and Singer (1999)) have defined their approaches as unsupervised, they are introduced as semi-supervised approaches, since an initial hand-crafted seed set is required. Applying semi-supervised learning can solve the coverage problem that is caused by the limitation of the rule-based methods and avoid the requirements of hand-labeled training data for supervised learning approaches. However, the recognition results can be sensitivity to the seed selection and the semantic drift problem can also occur during iterations.

Unsupervised learning

Alfonseca and Manandhar (2002) introduced an NER system, which, by adapting WordNet [Fellbaum (1998)], recognizes a named entity without requiring any labeled data. They first built a topic signature for every WordNet synset (e.g., county, man, and dwarf)

from the web sources following the strategy introduced by Agirre et al. (2000) based on the co-occurred word of the corresponding synset. Each named entity is assigned to the synset that maximizes the similarity score between the context of a named entity and a topic signature.

Elsner et al. (2009) reformulated the NER problem to the typical clustering problem. By clustering all named entities that are detected into different clusters, one can identify the type of named entities afterward. The researchers tested their system on the MUC-7 dataset and achieved 86% accuracy.

Discussion: Unsupervised learning methods avoid the requirements of labeled data. The idea is to transform the NER problem into clustering. The named entities are clustered according to their types. An unsupervised learning approach that does not exploit any language dependent feature can be applied for recognize named entities from text in different languages, since it does not rely on any supervision from humans. However, as mentioned by Elsner et al. (2009), the semantic information of clustering results is missing. Therefore, further processes are required to identify the represented entity types of each cluster.

2.2.4 Multilingual NER

Although the major focus of NER is on English text, a vast amount of work has been done to recognize named entities from texts in different languages. Due to distinct characters in various languages, it is difficult to find a unique solution; thus customized approaches are required. For example, the capitalization of the first letter typically indicates the mention of a named entity in English texts. This essential feature is widely used to detect the named entities in an English text. However, in the orthography for other languages, such as Chinese, Urdu, and Japanese, no similar feature is available. Even in some Roman languages, the capitalization feature is not as effectual as it is in English. For example, all nouns begin with a capital letter in German texts. Another example of languages with special challenges is Chinese. In the Chinese language, there is no clear indicator like empty spaces between different words in a sentence. Therefore, before solving the problem of NER, an extra preprocessing step is required to segment sentences into words.

Many researchers focus on solving the NER problem from specific languages other than English, including Chinese [Gao et al. (2005)], German [Ploch et al. (2012)], Spanish [Kozareva et al. (2005)], Hindi [Li and McCallum (2003)], Arabic [Benajiba et al. (2007)], Portuguese [Milidiú et al. (2007)], Bengali [Ekbal and Bandyopadhyay (2008)], Japanese [Isozaki and Kazawa (2002)], Malay [Alfred et al. (2013)], French [Petasis et al. (2001)], Urdu [Riaz (2010)], Greek [Farmakiotou et al. (2000)], and Korean [Chung et al. (2003)].

Some approaches are designed for multilingual named entity recognition, for instance, [Bikel et al. (1997); Florian et al. (2003); Kim et al. (2012)].

2.2.5 NER on short documents

With the extremely rapid growth of social media, especially the emergence of Twitter[1] in 2006, the task of NER from short texts became an interesting topic. Nowadays, millions of tweets are published every day. It is harder to extract named entities from social media data rather than from normal text documents, such as news articles, books, and scientific papers. Ritter et al. (2011) discussed the specific challenges of NER on tweets in detail. First, tweets contain a number of different named entity types, such as companies, products, and bands. However, most of the types have relatively low instances in tweets, so that even in a large corpus containing labeled tweets, the training examples for some entity types are still sparse. Second, tweets typically lack contexts since the maximal length of each tweet is limited to 140 characters. Third, most of the global features are not directly adaptable either. Furthermore, due to this limitation of the maximal length, users try to include as much information as possible in a short piece of text. Therefore, the rules of grammar are often violated and many new terms are created for saving spaces. For example, the word "tomorrow" can be written as "2m", "2morrow", "tmwr", and so on. Some example tweets are as follows:

1. *We're working on a GTA Online title update to address common issues on both platforms, We hope to have it out tmrw: rsg.ms/1dWHhLK*,
2. *Don't forget to tune in to @TheEllenShow to see @AriannaGrande perform #Problem today! youtu.be/VJ44m6-3ZLQ*, and
3. *@HouseofCards probably one of the most controversial scenes and best lines in tv-media history - last scene season 3 ep4.*

Starting from 2010, the topic of NER on Twitter data attracted much attention. Finin et al. (2010) used crowdsourcing platforms to annotate named entities in tweets into four categories, person, organization, location, and "none of the above". Liu et al. (2011) combined a KNN classifier and a CRF classifier to recognize ENAMEX and product names from tweets. Ritter et al. (2011) presented the T-NER system, which rebuilds the NLP pipeline from POS tagging to NER. The T-NER system can classify 10 different types of named entities including company, brand, product, movie, TV show, etc. Li et al. (2012) introduced the unsupervised NER system, TwiNER, that does not require any human label efforts is introduced. The TwiNER also employs Wikipedia as a source of *global context* to improve its performance.

[1] https://twitter.com/

Table 2.2: Experimental results of some NER approaches

Method	Approach	Dataset	F-measure
DT	Paliouras et al. (2000)	MUC-6	88.07% (ORG) 94.28% (PER)
HMM	Zhou and Su (2002)	MUC-6	96.6%
		MUC-7	94.1%
	Bikel et al. (1997)	MUC-7	93%
Rule-based	Fukumoto et al. (1998)	MUC-7	76.4%
	Mikheev et al. (1998)	MUC-7	92.29%
	Black et al. (1998)	MUC-7	92.5%
	Mikheev et al. (1999)	MUC-7	93,39%
EntMax	Borthwick (1999)	MUC-7	92.00%
	Chieu and Ng (2002)	MUC-6	93.27%
		MUC-7	87.24%
CRF	McCallum and Li (2003)	CoNLL-2003	84.04%
AdaBoost	Carreras et al. (2003)	CoNLL-2003	84.12%
Boostrapping+CRF	Liao and Veeramachaneni (2009)	CoNLL-2003	70.24% (ORG) 78.28% (PER) 79.22% (LOC)

2.2.6 Datasets

In 1996, the NER task was explicitly defined in MUC-6 [Grishman and Sundheim (1996)]. The dataset that contained labeled texts was provided for testing the performance of NER systems. In the dataset, 100 articles were drawn from approximately 58,000 Wall Street Journal articles. 30 out of the 100 articles were selected for testing. This is one of the earliest popular used test sets for the NER task. In the next year, MUC-7 [Chinchor and Robinson (1997)], provide a subset of the North American News Text Corpora. In the test set, around 800 named entities are annotated.

CoNLL-2003 shared task [Tjong Kim Sang and De Meulder (2003)] that provided a dataset that included training and test data in English and German. The goal of the task is to build up language-independent named entity recognition systems. The English data are generated from the Reuters Corpus[2], which contains one year Reuters articles from August 1996 to August 1997. The German data were extracted from the ECI Multilingual Text Corpus[3], which contains texts in many languages. The CoNLL-2003 dataset has to be further used for the NEL task with an additional hand-labeling effort.

These three datasets are frequently used to examine the performance of an NER system on English texts. The articles included in those datasets are all from news sources. Furthermore, as summarized by Augenstein et al. (2017), ACE [Walker et al. (2006)] and OntoNote [Hovy et al. (2006)] contain not only news articles but also several

[2] http://about.reuters.com/researchandstandards/corpus/
[3] https://www.ldc.upenn.edu/

sub-corpora such as broadcast conversation, web logs, and telephone conversations.

Nothman et al. (2013) automatically created a multilingual annotated dataset for NER. They exploited the text and structure of Wikipedia and generated five annotated training corpora in different languages including English, German, Spanish, Dutch, and Russian. Each of the corpora contains 3.5 million tokens, which are labeled according to four classes: person, organization, location, and miscellaneous (MISC).

In the biological domain, Kim et al. (2003) generated a corpus called GENIA, which consists of 2000 MEDLINE abstracts. Out of more than 400,000 words, almost 100,000 biological terms were annotated as named entities.

Table 2.2 lists some of the approaches that are introduced in the previous subsections. Please note that, this table includes only the approaches that reported their evaluation results on the datasets, MUC-6, MUC-7, or CoNLL-2003 in the original papers. As the evaluation results show, many approaches can generate promising NER results on the benchmarking datasets.

2.2.7 Conclusion

The task of NER is well-defined and researched and attracted researchers attention for approximately 20 years; much work has been done during this long period. As the fundamental step of a named entity mining system, the quality of NER results propagate to the final result of the named entity mining system. With the development of the NER system, many approaches can achieve NER results that are comparable to human annotations, especially recognition of the general entity types from English texts. Some of the studies also focus on NER from texts in other languages and some specific domains, e.g., the biological domains have received much attention. The appearance of new data types can bring new challenges. For example, after the explosion in the volume of social media data that contains a vast amount of information, recognizing named entities from short texts became a popular topic.

2.3 Named entity linking

The task of NER aims to only identify mentions of named entities and their types. In this section, we present an overview of various aspects of the named entity linking task, which focuses on disambiguating mentions of named entities. We first introduce several commonly used knowledge bases in detail. In the following parts, we present the general procedure of NEL systems: candidate entity extraction, candidate entity ranking, and recognition of entities that are not in knowledge base. Finally, we shortly introduce the related work on NEL from social media data and several commonly used benchmark datasets.

2.3.1 Knowledge bases

According to the definition of the NEL task, which was introduced in Section 1.2, a knowledge base is one of the most critical components. In general, a knowledge base is a repository of information. It is a library of knowledge regarding a particular topic. It is worth noting that the popular large-scale KBs typically attempt to cover as many entities as possible. In such a knowledge base, not only the definition of entities but also a vast amount of information is included, which can be used for disambiguating mentions of name entities.

Knowledge bases, such as Wikipedia[4], YAGO [Suchanek et al. (2007)], DBpedia [Auer et al. (2007)], Freebase [Bollacker et al. (2008)], and Wikidata [Vrandečić and Krötzsch (2014)] are widely used for NEL. Especially, with the growth Wikipedia, the contained information became one of most important sources for building up a knowledge base.

Wikipedia

Wikipedia is a free online encyclopedia that includes millions of articles in 295 languages. Wikipedia allows users to edit articles freely; many domain experts also participate in enriching and improving Wikipedia articles. Since the first launching of Wikipedia in early 2011, the number of articles on Wikipedia increases rapidly. The total amount of English articles in Wikipedia increased from only around $20,000$ to over $5,000,000$ in 15 years as shown in Figure 2.1. Until 2017, articles in Cebuano, Swedish, and German languages exceeded two million. Wikipedia contains articles of different types including general entities, named entities, historical events, etc.

Wikipedia uses an article's title as its identification, meaning that each article has a unique title. When a name is ambiguous, it is typically extended by some expressions according to its unique character. For example, the article about the apple is named *Apple*, the one about Apple, the technology company, is named *Apple Inc.*, and the one that introduces the British rock band called Apple is named *Apple (band)*. In the main body of a Wikipedia page is content that introduces the relevant topic. Every Wikipedia page is assigned to multiple category labels. For example, the page about Steven Jobs belongs to categories including *1955 births*, *Technicians*, *People from Cupertino, California*, and so on. Besides, some of the pages include infoboxes, which contain semi-structured relevant information about their subjects.

Wikipedia is one of the most important data sources for named entity linking; as we have mentioned, many knowledge bases are generated based on it. Besides, Wikipedia itself has also been used as a knowledge base for the task of NEL. The information that is contained in Wikipedia pages is also useful for solving the NEL problem. Some useful features or particular types of pages in Wikipedia can be later used for NEL:

[4] www.wikipedia.org

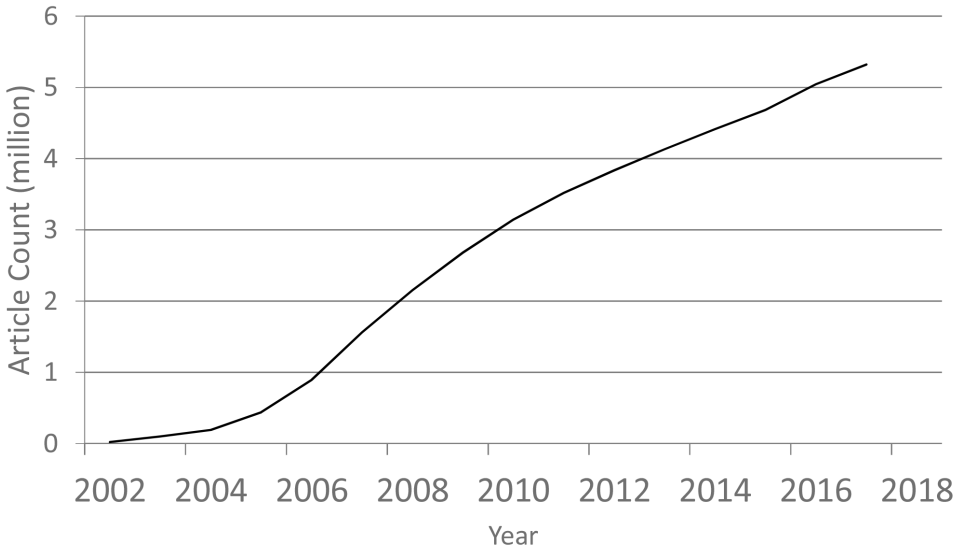


Figure 2.1: Changes in the number of English articles on Wikipedia. (Source from: https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia)

Infobox: In many Wikipedia pages, a summary of some unifying aspects that relate to the topic of the article is presented in the form of an *Infobox*. For different types of entities, attributes included in an infobox can be different. For example, the Wikipedia page of a company could include attributes such as company type, headquarter, founders, founded time, area served, and number of employees. Based on the representation format of infoboxes, it is relatively easy to gain some key information from them.

Hyperlinks: They are another useful source contained in Wikipedia articles. Authors are required to link at least the first mentions of other Wikipedia entities to the corresponding articles. An example sentence in Wikipedia articles would be "*... in Germany's post-war auto market, sandwiched between [[Volkswagen—VW]] and [[Mercedes-Benz]]*". In the original sentence, "VW" is mentioned and hyperlinked to the Wikipedia page of the company *Volkswagen*, and the company *Mercedes-Benz* is mentioned by its Wikipedia page title. These hyperlinks are valuable for extracting the correlations between different Wikipedia pages.

Categories: Each of Wikipedia pages is assigned to multiples category labels, which can be considered to be the topics to which the corresponding entity belongs. For example, the page about *Apple Inc.* is assigned to categories such as *Computer companies of the United States*, *Apple Inc.*, and *Steve Jobs*. An interesting example is that Apple also belongs to the category of *Steve Jobs*, since Apple is closely related to Steve Jobs. Other Wikipedia pages, such as the company *Pixar*, the film named Steve Jobs, the Jacking

house that was owned by Jobs, also have the category label *Steve Jobs*. The categories information provides direct clustering of Wikipedia pages. However, Wikipedia categories do not form taxonomy with summation hierarchy. Therefore, due to this complexity, information also needs to carefully preprocessed.

Category pages: Besides the category labels, Wikipedia contains category pages. A category page typically contains a list of page titles that belong to the corresponding category and a list of subcategory pages. For example, in the category page "Category: Capitals in Europe", the pages of capital cities in Europe are listed, for example, Berlin, Paris, Amsterdam, and so on. With this list, one can easily derive the information that all listed cities are capitals of European countries.

Disambiguation pages: For each ambiguous name, there is a specific type of Wikipedia page, called the disambiguation page. Each of the pages contains all the possible entities to which the corresponding name can refer. It is one of the most convenient ways of extracting the possible candidate entities by mentioning a named entity. As in the previous example, the disambiguation page of Apple includes around 50 unique named entities that belong to different domains, such as companies, music, and places.

Redirect pages: Opposite to the disambiguation pages, the title of a redirect page can be considered to be an alias name that refers to a Wikipedia entity. For instance, since "VW" is a popular abbreviation of the German automotive company Volkswagen, the page titled *VW* is a Wikipedia redirect page that is a pointer to the Wikipedia article *Volkswagen*. In most of the cases, the title of a redirect page is a typical alias name of the redirected one.

YAGO

YAGO [Suchanek et al. (2007)] is a state-of-the-art knowledge base that was been published in 2007.In the first version, it contains over one million entities and five million facts that are automatically extracted from Wikipedia articles and integrated with WordNet Synsets. Except for the ontological facts, there is a specific relation context, which contains about 40 million instances. The relation context is defined as linkages (i.e., hyperlinks) in one Wikipedia article to other Wikipedia pages.

In 2011, Hoffart et al. (2011a) presented YAGO2, which further extended YAGO. After including temporal and spatial knowledge from Wikipedia, GeoNames, and WordNet, YAGO2 now contains nearly 10 million entities and 80 million facts. Regarding to temporal dimension, two universal relations startsExistingOnDate and endsExistingOnDate are defined to be an entity's temporal start and end points. For different types of entities, the underlying meaning of these two relations is variant. For example, they

indicate the wasBornOnDate and diedOnDate relation for the entity type people, but the startedOnDate and endedOnDate for events. In YAGO2, a new class yagoGeoEntity consists of a physical location by which an entity is defined. For this specific class, the hasGeocCoordinates relation (i.e., the geographical latitude and longitude) is extracted for each location entity.

Suchanek et al. (2013) re-factored the architecture of YAGO, which is called YAGO2s. After that, Mahdisoltani et al. (2014) released the latest version of YAGO and further extended YAGO by combining the information from Wikipedia in 10 different languages, including European languages (e.g., German, French, and Dutch) and non-European ones (i.e., Arabic and Persian).

DBpedia

DBpedia [Auer et al. (2007)] was introduced in 2007 and consists of structured information extracted from Wikipedia. It uses Wikipedia dumps that are extracted by MediaWiki[5]. In these dumps, Wikipedia articles and some structured information that can be directly extracted by MediaWiki are stored in a relational database. Then, DBpedia processes these dumps to collect semantic relations from the structured information or extracting them from texts.

The first version of DBpedia consisted 1.95 million entities in types such as person names, locations, music albums, and films. These entities are further linked to the relevant images, external web pages, YAGO categories, and so on. DBpedia also includes abstract descriptions for most entities in 13 different languages. There are already around 103 million RDF triples in this version.

With the inclusion of more features and the rapid growth of Wikipedia, DBpedia contains more and more valuable information. As reported by Bizer et al. (2009), after two years, DBpedia covers over 2.6 million entities, while the descriptions are extended to 30 languages. At the same time, nearly 4.7 billion RDF triples are included. In the latest publication, Lehmann et al. (2015) introduced the actuality of DBpedia. After eight years developments of the first release, DBpedia has extracted information from 111 languages, among which the largest one is the English version. It consists of over 400 million facts of 3.7 million entities.

Freebase

Freebase [Bollacker et al. (2008)] was another well-known knowledge base consisting of data that are composed mainly of its community members. It was initially developed by Metaweb, which was later acquired by Google in 2010. The information is collected from

[5] https://www.mediawiki.org/

not only Wikipedia but also other sources such as *Notable Names Database (NNDB)*[6], Fashion Model Directory[7], and MusicBrainz[8]. It contains more than 125 million tuples in over 4,000 types. On May 2, 2016, Freebase was officially shut down, and part of its data was moved to Wikidata [Pellissier Tanon et al. (2016)].

Wikidata

In 2014, Vrandečić and Krötzsch (2014) introduced an open editable multilingual system, called Wikidata. It was designed for manage Wikipedia data in a better way. For example, Wikidata solves the conflicts that exist among different language versions of the same Wikipedia entity by organizing them in a plurality manner. The data in Wikidata are stored in different formats such as JSON and RDF to ensure that they are easily accessible.

Table 2.3: Overview of KBs introduced by Färber et al. (2016).

	YAGO3	DBpedia*	Freebase*	Wikidata*
#Entity Classes	569,751	736	53,092	302,280
#Instances	12,291,250	20,764,283	115,880,761	142,213,806
#Named Entities	5,130,031	4,298,433	49,947,799	18,697,897
#Relation Types	106	60,231	70,902	1,874
#Relations	1,001,461,792	411,885,960	3,124,791,156	745,530,833

In summary, Färber et al. (2016) analyzed and compared several KBs in detail. Table 2.3 shows part of statical analysis results presented in their work. Please note that the version of the KBs for comparison are DBpedia (April 2015), Freebase (March 2015), and Wikidata (October 2015). In this table, entity classes refer to the distinct type of all instances (i.e., general entities) in each KB. Due to the fact that the definitions of the same element in different KBs are not consolidated, this result only presents a high-level overview. For example, in Freebase, each direction of a type of an asymmetric relation is considered to be a unique type.

Please note that these popular KBs include not only named entities but also general entities. For example, a Wikipedia page can also be a general entity (e.g., tree, chair, and water), an event, a list of a group of items, and so on. Therefore, for the NEL task, a selected KB is typically preprocessed to keep only the named entities for a linking task.

[6] http://www.nndb.com
[7] http://www.fashionmodeldirectory.com/
[8] http://musicbrainz.org/

2.3.2 Candidate entity selection

As introduced in Section 1.2, the candidate selection step is the first phase of NEL. In this step, each mention is assigned with a candidate list that contains all possible entities to which the mention can refer. With candidate entity lists, an NEL system can focus on establishing the correct linkages from only the reasonable candidates in the next step. The quality of this candidates generation step is crucial for the entire NEL system. On the one hand, a candidate list should cover all possible entities to make sure that the correct one is included. Otherwise, the system can never give out the correct result regardless of how the candidates are ranked. On the other hand, it is also important to keep the list as short as possible to shrink the search space; this can improve the performance of the NEL system in terms of quality and efficiency.

Different techniques are applied to extract candidates. However, in summary, the commonly applied strategy is to build up a dictionary that maps each mention to all name entities in KBs to which the mention can refer. Two sources (i.e., KBs and surface forms) are frequently used for extracting candidates.

KB-based methods

For the NEL task, KB is an important data source for generating a candidate list. As the purpose of NEL, mentions need to be linked to entities in KB. Directly extracting candidates based on the information in KB is the same as establishing preliminary linkages between mentions and entities. The structural features and some further information in KBs can be used not only for candidate extraction but also for the following ranking stage.

Wikipedia is a useful source from which to extract candidates. As mentioned in Section 2.3.1, some pages or structures in Wikipedia, such as redirect pages, disambiguation pages, and hyperlinks are valuable to building up a dictionary. Bunescu and Pasca (2006) selected out pages about named entities according to some rules,. For each named entity, the titles of the original page as well as corresponding redirect and disambiguation pages are stored as entries in a dictionary. After this process, each entry in the dictionary matches to one or more named entities. Cucerzan (2007) further included the hyperlinks from Wikipedia articles and built up a dictionary consisting of more than 1.4 million named entities and 3.36 million surface forms (i.e., entries).

Similarly, page titles, redirects and disambiguations in DBpedia are used to build up a dictionary by Mendes et al. (2011a). Hoffart et al. (2011b) harnessed the *means* relation in YAGO to generate candidate lists.

Mention-based methods

Although KBs cover most of the various alias forms of a named entity, some of the possible surface forms can be missing.

Gottipati and Jiang (2011) expanded mentions of named entities according to the given documents. For example, a mention of the person name "Sophia Coppola" can be further expanded to be "Coppola" and "Sofia Coppola". Jain et al. (2007) designed an approach to expanding acronyms from different sources on the Web. They ranked the expansions according to their ranking strategy, which includes popularity and reliability factors. For instance, mention "ABS" can refer to American Bonsai Society, American Bamboo Society, Atlas Business Solutions, and so on.

2.3.3 Candidate entity ranking

Candidate ranking is the core procedure of disambiguating mentions of named entities. With variant disambiguation strategies, entities in a candidate list are ranked according to the probability of a given mention to refer to each of them.

Ranking features

In this subsection, we introduce prominent features for the NEL task. Following the definition of NEL, the core of the task is to find a named entity in a KB that is most similar to a mention. Contexts of a mention and knowledge about a named entity in a KB are the two primary sources of measuring the similarity. Therefore, we summarize the frequently exploited features in two categories according to the extraction sources: *context features* and *knowledge base features*.

Context features: The forthright strategy of getting clues for NEL is from input texts or textual data in KBs. Both mentions of a named entity and the surrounding context are primary sources for the NEL task.

Surface: A Surface (i.e., a mention) is a collection of terms that are used to mention named entity in texts. The string similarity between a mention and a candidate entity can be used as direct information for NEL. The approach introduced by Zheng et al. (2010) employs a group of surface features (eight features) for candidate ranking. A surface is compared with each of its candidate entities as follows: (1) the edit distance value (e.g., the Levenshtein distance [Navarro (2001)]), (2) whether the candidate and the surface are the same, or the surface is the prefix or postfix of the candidate, and vice versa, and (3) the number of common words and different words between them.

Furthermore, Christen (2006) introduced the *longest common sub-string* measure in

the person name matching field. Based on that, Dredze et al. (2010) included the ratio of the longest common sub-string as a feature. Liu et al. (2013) redefined edit distance similarity to capture the abbreviated forms of named entities in tweets. For example, "MS" can be an abbreviation of "Microsoft" since the length of "MS" plus the edit distance between them is equals to the length of "Microsoft".

Textual Context: The contextual similarity is one of the most important features for NEL. For an ambiguous mention, context gives the direct clue of a named entity to which a mention refers. Many NEL approaches use the contextual similarity between the context of a mention and the description text of a candidate entity (e.g., a Wikipedia article) [Cucerzan (2007); Hoffart et al. (2011b, 2012)].

Texts need to be converted into vector presentations for calculating textual context similarities. For example, Bagga and Baldwin (1998), Mann and Yarowsky (2003), as weel as Dredze et al. (2010) applied the bag of words method. The idea is to formulate the entire document of snippets in certain ranges as a set of words or phrases. Considering the fact that not all words in contexts are valuable for disambiguating a named entity, the words or phrases are typically assigned with different weights to capture their importance to a corresponding entity. *Term Frequency-Inverse Document Frequency (tf-idf)* is a good indicator for the importance. For example, Fader et al. (2009) and Mann and Yarowsky (2003) used the tf-idf score as the weight factor for each word. Mendes et al. (2011b) further introduced *Inverse Candidate Frequency (icf)* weight. Comparing to the idf weight brings the view of the whole corpus, icf weights words based on their ability to distinguish between the candidates. Inverse candidate frequency is defined as follows:

$$ICF(w_j) = log\frac{|R_s|}{n(w_j)} \tag{2.1}$$

where R_s refers to the set of resources for the candidates of mention s, and $n(w_j)$ is the number of resources contains w_j. This tf-icf weight has been used by Mendes et al. (2011a) for NEL.

With these vector representations, there are various ways of calculating the contextual similarities between vectors, including the dot product, the cosine similarity, Dice's coefficient, and so on.

Knowledge base features: In the NEL task, a natural source that contains rich information about named entities is the aimed knowledge base. Except for the textual description of each entity, some structural information contained or derived from a KB is valuable for candidate ranking. Wikipedia is such a source for extracting features for disambiguation. As mentioned earlier, redirect pages, disambiguation pages, and hyperlinks in Wikipedia are helpful for candidate ranking.

Category information: In Wikipedia, each page (i.e., entity) is assigned by several

category tags. With these category tags, one can easily separate the Wikipedia entities into different groups. For example, Bunescu and Pasca (2006) employed the category information as taxonomy kernel. Cucerzan (2007) extracted category information from special pages whose titles start with "List of" or "Table of".

Prominence information: The Wikipedia hyperlinks contain both mentions and linked entities. After summarizing the frequency of the linkages from many hyperlinks, we know that the probability of each unique mention refers to each entity. For example, according to this probability the top-3 candidate entities for the mention "Jordan" are the country Jordan, Formula One constructor Jordan Grand Prix, and the American ichthyologist David Starr Jordan. This popular feature is frequently used in many NEL approaches, e.g., [Guo et al. (2013a); Hoffart et al. (2011b); Zuo et al. (2014)]. Instead of using hyperlinks, Dredze et al. (2010) employed Google's PageRank algorithm to measure popularities of Wikipedia pages.

Entity types: An ambiguous mention can refer to different types of named entities. In the previously mentioned example, the mention Jordan could refer to a location, an organization, a person, etc. In this case, by including information on the entity type, a system can focus on disambiguating only the candidate entities with the correct type. To get this information, one can apply a reliable NER approach to recognize entity types of mentions. For entities in KBs, there is typically structural information available. Dredze et al. (2010) applied this strategy. For instance, if the mention "John F. Kennedy" has already been recognized as a person, it should refer to a person name. However, there is a clear disadvantage of this feature. It highly relies on the performance of the employed NER approach.

Relation information: The relations among entities are also useful for candidate ranking. Han and Zhao (2009) constructed a semantic network based on redirect and disambiguation pages as well as hyperlinks in articles, so that their system can consider the semantic relatedness when measuring similarities. To achieve this purpose, the researchers used the method described by Witten and Milne (2008) to measure the semantic relatedness between two Wikipedia concepts. For example, the semantic relatedness between *NBA* and *Bayesian network* equals to zero, but the relatedness between *Machine learning* and *Bayesian network* equals to 0.74. Gruetze et al. (2016) captured coherence through the random walks algorithm. Coherence is defined to be the probability that a random walker starts from one candidate entity to another following hyperlinks in Wikipedia.

Ranking techniques

State-of-the-art NEL systems employ a mixture of variant features for candidate ranking. The common idea is to rank named entities in a candidate list according to the similarity between the mention and each of the candidates. With the ranked candidate list, the mention is linked to the most similar candidate. According to the ranking methodologies, we introduce some commonly used ranking strategies in three categories, which are

vector-based similarity, *probabilistic model*, and *relation-based similarity* strategies

Vector-based methods: In traditional methods, each mention and each named entity are represented by a vector of terms that are in their textual context. Vector-based similarity measures are applied to capture the affinity between a mention and a named entity. The feature values can go beyond simple unigram terms and consist of compound terms, such as bigrams, key phrases, encyclopedic facts, and categorical descriptions. For example, Bagga and Baldwin (1998) used the *Vector Space Model (VSM)* to disambiguate person names from different documents. A summary is generated for each document with respect to the mention of interest. VSM is used to compute similarities between summary pairs and the pairs with similarity scores greater than a threshold are considered to describe the same entity. Mann and Yarowsky (2003) included biographical information, such as date of birth, place of birth, and occupation, in the vector representation of a named entity. Pedersen et al. (2005) employed salient bigrams to represent the context of a mention. Bunescu and Pasca (2006), after deriving an entity dictionary from Wikipedia, for a given mention, ranked entities by a SVM kernel-based similarity between the textual context of the mention and the Wikipedia text as well as categories of the candidate entity. The mention is linked to the most similar entity.

Cucerzan (2007) also employed the VSM method for disambiguating named entities. In his work, articles and category tags of named entities are presented in a vector form. The context of each mention in a given document is also transformed into vectors. The system linearly combines the contextual similarity (i.e., the vector similarities between the contexts of candidate entities and entity mentions) and the similarity of category tags (i.e., the agreement between any two selected entities in the document).

Fader et al. (2009) introduced the GROUNDER system, which links a named entity to Wikipedia entities. The GROUNDER system converts given documents and Wikipedia articles into tf-idf vectors and computes the cosine similarity score between the vectors. In addition, GROUNDER employs the lucene-search[9] extension from MediaWiki to compute a prior score of a name entity in a given query (i.e., context). The system links a mention to the candidate entity that maximizes the overall score that is calculated by multiplying the previous two scores.

Probabilistic methods: The ranking problem has also been formulated as a probabilistic reasoning problem. Fleischman and Hovy (2004) trained a maximum entropy model to infer the probability that two mentions represent the same entity. In this work, they focused on the person name resolution problem, which can also be considered as a named entity disambiguation problem. The approach also used a modified agglomerative clustering algorithm to cluster mentions of person using the probabilistic similarity measure. Similarly, Sil et al. (2012) used a log-linear model to represent the probability of a named

[9] https://www.mediawiki.org/wiki/Extension:Lucene-search

entity to which a mention refers. For both of the methods, the selection of features and efficient strategies for learning their weights are crucial, as ideally, all feature weights should be learned in a joint fashion, which can be computationally expensive and is often impeded by dimensionality. Mihalcea and Csomai (2007) introduced the Wikify! system to link general entities to Wikipedia pages. The disambiguation strategy is based on a Naive Bayes classifier, which was introduced by Mihalcea (2007). Candidate entities are ranked according to the probabilities that a mention refers to them. Please note that, in this work, the researchers focused on the task of word sense disambiguation, which also disambiguates normal nouns comparing with NED. Demartini et al. (2012) introduced the ZenCrowd system, which links mentions to *Linked Open Data (LOD)* by exploiting probabilistic networks. Except for the automatic process, ZenCrowd dynamically leverages human intelligence feedbacks (i.e., crowdsourcing).

Relation-based methods: As introduced above, relation features can be extracted from KBs. These features has been proved to be a valuable feature for the disambiguation process. Du et al. (2013) employed similarity measures that captured the average pair-wise proximity between candidate entities in the knowledge graph and their average pair-wise conceptual similarity by means of the lowest-common-ancestor classes. Hoffart et al. (2011b, 2012) exploited the hypernymy- and key-phrase-based relatedness between k candidate entities in the knowledge base to jointly link k mentions that occurred in the same paragraph. A prior probability of a candidate entity to be referred to by a mention was combined with the above relatedness measures in an objective maximization function. The intuition behind the hypernymy-based relatedness was that for k mentions (that occur in the same textual context) to be linked correctly to lk named entities in the knowledge base, the l entities should jointly exhibit a high semantic relatedness, which in [Hoffart et al. (2011b)] is referred to as coherence the AIDA approach. Despite this principled modeling of the NEL problem in AIDA [Hoffart et al. (2011b)] and KORE [Hoffart et al. (2012)] approaches as well as the impressive quality results reported in those works, efficiency seems to be the main bottleneck of such collective inference models.

Discussion: The three introduced strategies are often applied to solve NEL problem. They can also been combined in one approach for ranking candidate entities. Vector-based methods disambiguate named entities by comparing the similarity between contexts of mention and description documents of named entities directly. It often performs well in terms of runtime, since the main process is to calculate distances between vectors. However, the context of a mention does not always match the description text of the referred named entity. Especially when some keywords for disambiguating a mention are missing, a wrong linkage can be established. Furthermore, some information, e.g., the order of the terms appear in a document, is lost when transforming terms into vectors. With a probabilistic method, entities in a candidate list can be ranked according to the probabilities of the mention refers to them. The weights of exploited features

can be learned and fine-tuned to improve accuracy performance. But it also can be expensive to find out the proper weights for each feature. Besides, exploited features need to carefully selected and extracted. Finally, by applying relation-based similarity strategies, approaches take the relatedness information among mentioned named entities into account. This strategy can improve disambiguation process when mentioned named entities are tightly related to each other. However, the runtime cost is often expensive, especially when the candidate space is large. It can also mislead the disambiguation process when a mention refers to the candidate entity that is not the most relevant candidate to other named entities in the same query. For example, if the main topic of a document is the football clubs in Germany, most of the mentioned named entities in this documents are club names. A mention, which actually refers to a city name, can be wrongly linked to the football club in that city.

NIL entity prediction

Although the popular KBs contain a vast amount of entities, there is still a chance that some entities are missing (i.e., NIL entities). For this case, a good NEL system should link the corresponding mention to an NIL entity.

According to the general framework design of NEL systems, the candidate entity selection step is the first filter of NIL entity mentions. The intuition is that if no candidate can be generated for a certain mention, the correct named entity is not included in the KB.

In the candidate entity ranking step, most of the NEL systems rank a candidate according to some similarity scoring function. To recognize NIL entities, a threshold is typically involved. For example, in [Bunescu and Pasca (2006); Lehmann et al. (2010); Shen et al. (2012)], the score of the top-ranked candidate need to exceed a given threshold; otherwise, the mention is considered to be referring to a named entity not present in KB. Except hand-tuning a threshold, Dredze et al. (2010) included a NIL entity as a candidate for each mention. A mention is linked to NIL when NIL is top-ranked.

Monahan et al. (2011) trained a binary logistic classifier to measure the likelihood of a top-ranked candidate to be a correct result. The mention was linked to NIL when the classifier rejected the top-ranked candidate.

The prediction of NIL entities can improve the quality of NEL results by avoiding linking unknown named entities to the most similar one in KB. Furthermore, the result of NIL entity prediction can also be used to augment KB. Ploch et al. (2011) employed hierarchical-agglomerative clustering algorithm to cluster identified NIL entities. They implemented a three-stage clustering approach, which clusters mentions according to three forms, surface form, ambiguous surface form, and synonymous form. Combining the clustering result with document similarities, their approach can further disambiguate mentions of NIL entities and suggest valid entries to augment KB.

2.3.4 NEL on short documents

Similarly to the NER task, social media data such as tweets also bring new challenges for the NEL task. Two main challenges are the lack of available contextual information and the rich variations of mentions.

Meij et al. (2012) focused on linking tweets to relevant Wikipedia pages. Except for general features, they introduced new tweet features. First, a tweet is compared to Wikipedia titles to see whether they are the same or sequences of one another. Then, Tweets were further expanded by the corresponding webpages when they contained some hyperlinks and other tweets with the same hashtag. Liu et al. (2013) designed an NEL framework using a collective inference method to simultaneously link a set of mentions by integrating mention-entry, entry-entry, and mention-mention similarity. Some other studies were conducted for NEL on tweets [Gattani et al. (2013); Guo et al. (2013a,b); Yang and Chang (2016)].

2.3.5 Datasets

Starting from 2009, NEL has been introduced as a subtask of the *Knowledge Base Population (KBP)* track at the *NIST Text Analysis Conference (TAC)*. Benchmark data sets were provided for evaluating NEL systems. The most regularly used ones are TAC-KBP 2009 and 2010.

The TAC-KBP 2009 [McNamee and Dang (2009)] corpus contains 3,904 mentions of 560 distinct named entities, including 15% Person (PER), 70% Organization (ORG), and 15% Geopolitical Entity (GPE). Compared to the Wikipedia version on October 2008, 67.5% of the mentions are not included (i.e., NIL entities). In the following year, TAC-KBP 2010 [Ji et al. (2010)] contained 2,250 test mentions, of which only 28.4% were NIL.

As introduced by Shen et al. (2015), a problem in the TAC-KBP dataset is that most of the documents contain only one mention. For example, in TAC-KBP 2009, there are 3,688 documents but only 3,904 mentions in total. Therefore, most of the systems including coherence features or other global features cannot be evaluated over these data sets. For evaluation, in some works, authors build up their own test set.

Cucerzan (2007) randomly selected 350 Wikipedia articles to build up a test set. In this dataset, 5,812 mentions are included and 551 of them do not refer to any Wikipedia article. Fader et al. (2009) collected 500 tuples and their relevant webpages based on the TEXTRUNNER system [Banko et al. (2007)]. Hoffart et al. (2011b) took the dataset for the NER task, i.e., CoNLL 2003 dataset [Tjong Kim Sang and De Meulder (2003)], and hand labeled the named entities with corresponding entities in YAGO2. The new dataset contains 34,956 mentions in 1,393 news articles.

Various knowledge bases can be used by the state-of-the-art NEL approaches. Therefore, many researchers created their own gold standard datasets for evaluating their approaches. It is difficult to directly compare the performances of the introduced approaches according to the reported results in the original work. In Section 3.5, we applied our NEL approach and other three approaches developed by Cucerzan (2007); Hoffart et al. (2011b, 2012) on three different datasets and presented the evaluation results.

2.3.6 Conclusion

As we have introduced above, the major task of NEL is to disambiguate the mentions of the named entities. Therefore, researchers mainly focus on improving the accuracy of the disambiguation processes. This is a challenging task due to the ambiguity of mentions in texts. A straightforward source for feature extraction is from the given text, which is the mention itself and the surrounding contexts. However, the key information for disambiguation is typically hidden in contexts, which need to be captured by some techniques. Even worse, sometimes the information is difficult to match to that in KBs or is even not included. Therefore, other features are often required to improve performances of NEL systems. As a crucial component of the NEL task, knowledge derived from the aimed knowledge bases can be used to solve the NEL problem. For example, the category information, prominence feature, and coherence among entities are proved to be effective for solving the NEL problem.

2.4 Relation extraction

In this section, we introduce the state-of-the-art of relation extraction approaches. We mainly focus on introducing major aspects of the binary relation extraction task. We first present relation extraction approaches in detail including extraction features, techniques, and the widely used datasets. Thereafter, we briefly introduce the complex (i.e., n-ary) relation extraction task.

2.4.1 Extraction features

For the RE task, the primary information about relations is contained in contexts. Therefore, most RE approaches focus on capturing valid information that describes relations. First, the terms in contexts provide direct clues for relation extraction. Second, results of some NLP tasks are also useful features that are frequently exploited. Third, many approaches are developed based on lexico-syntactic patterns to solve the RE problem.

Term features

The terms in the context of two mentioned entities give the direct clues of the probable relations between them. In general, according to the occurrence positions, terms are separated into four categories: 1) mentions of both named entities, 2) terms in between two mentions, 3) terms before the first mention, and 4) terms after the second mention.

Terms in context can be directly used as numerous features, for example, the bag of words in different categories introduced above, the head term of each mention, and the first term before the first mention.

Some features that are derived from terms are also useful for solving the RE task. For instance, one can calculate the distance between two mentions with the number of terms in between.

Furthermore, another regularly exploited feature is the overlaps between two mentions. When one mention includes another one, or they contain some common terms, it is more probable that they relate to each other. Given two mentions, "Walt Disney Studios" and "Walt Disney Company", as an example, *Walt Disney Studios* is indeed one the subsidiaries of *Walt Disney Company*.

Furthermore, another regularly exploited feature is the overlaps between two mentions. When one mention includes another one or the mentions contain some common terms, it is more probable that they relate to each other. Given two mentions "Walt Disney Studios" and "Walt Disney Company" as an example, *Walt Disney Studios* is indeed one of the subsidiaries of *Walt Disney Company*.

NLP features

As a task in the NLP domain, RE approaches can benefit from the results of other NLP tasks, especially some fundamental ones. For example, with the information derived from POS tagging and NER, more features from the context can be extracted. Although these features are useful for solving the RE task, one notable limitation is that their performances highly rely on that of the exploited NLP approaches. For example, even though many existing NER approaches can produce high-quality results (e.g., over 90% in F-measure), wrong labels are still included; these can mislead RE approaches.

POS tagging: The result of POS tagging is an essential source for deriving features for the RE task. First, a straightforward feature is using POS tag of each term directly. Second, one can generate a parse tree using POS tagging. Furthermore, a POS tag is a major component in creating lexico-syntactic patterns, which are introduced below.

Parse tree: A syntactic parse tree is the representation of the grammatical structure of a sentence. With this information, some approaches use the path connecting two mentions in the parse tree as a feature.

From the full parse tree, base phase chunking can also be extracted as a feature. The base phase chunking segments a sentence into constituents such as noun, verb, and prepositional phrase. The information of base phase chunking is used to replace terms and create new features.

A dependency tree can be generated based on the dependencies information contained in a parse tree. Then the dependency terms of mentions are also considered to be keywords that might indicate the relations between mentions.

Entity type: By applying NER approaches, one can label the types of named entities in a given document. Since the RE task is aimed at finding out the relations between named entities, the type of a mentioned named entity is typically a determinant feature. For instance, for us to determine a persons birthplace, the two entities in this type of relation must be a location name and a person name. In this case, all combinations of other types of named entities can be directly abandoned for this particular relation. Furthermore, together with the term features, based on NER results, another feature is the number of other named entities between two mentions.

Lexico-syntactic patterns

Lexico-syntactic patterns are widely used in RE, especially in weak-supervised learning approaches (e.g., distant-supervised or semi-supervised learning). These patterns can be manually defined or automatically generated. For example, with a pattern "NP_1 is the capital of NP_2", one can extract the capital_of relation between Berlin and Germany from a snippet "*Berlin is the capital of Germany.*"

With the result of NER, one can define alternative patterns to extract relations. For instance, suppose the target is to find out a companys headquarters. Patterns such as "<COMPANY>'s headquarters in <LOCATION>" and "<LOCATION>-based <COMPANY>", are useful in finding the type of relations.

Patterns are often applied for extracting relations by matching texts directly. This is an efficient solution without any complicated processing, which can extract relations from a large-sized corpus. Moreover, the result generated using patterns is easily understandable by humans. However, the disadvantage of exploiting patterns is significant, which is the low flexibility. A single type of relation can be described in various ways. Relation instances cannot be extracted when their contexts do not match any of the selected patterns. It is the reason why the approaches that use only the pattern-matching strategy have limited recall performances.

2.4.2 Extraction techniques

State-of-the-art RE approaches apply different learning strategies for extracting relations between named entities. From supervised to unsupervised learning techniques, the requirements of labeled training data decreases. Since the distant-supervised learning strategy is often used for building up an RE system, we introduce it in a separate section.

Supervised learning

In supervised learning based approaches, the RE problem is reformulated as a classification problem, namely classifying whether two entities participate in a specific relation.

Girju et al. (2003) introduced an RE system to extract its partwhole relations, which is defined in WordNet, including member-of, staff-of, and part-of relations. Their system first extracts patterns based on selected WordNet pairs from a large corpus and then discovers semantic constraints for all patterns by training a decision tree model.

Kambhatla (2004) employed maximum entropy models to solve the RE task. In his approach, features including words, entity types, parse tree, and so on are extracted to build up a feature stream for each pair of mentions. He reformulated the RE task to be a multi-class classification problem. The approach classifies each pair of mentions into one of 49 relation classes (including a "NONE" class).

GuoDong et al. (2005) applied a feature-based relation extraction strategy that uses SVM. Their approach employs multiple binary SVM classifiers to separate each class from all other classes. The relation between a mention pair is determined by the class that maximizes SVM output. Moreover, apart from some basic features, their system includes base phrase chunking as an essential feature.

Kernel methods have been successfully applied to deal with the RE problem. The basic idea is to determine the relation between two mentions according to the similarity between their context and the labeled training sentences, which describe some targeted relation types. Zelenko et al. (2003) introduced a tree-kernels-based approach, in which sentences in texts are represented by shallow parse trees, which only identify the key elements in contrast to normal parse trees. Then, kernels enumerate all subtrees of two parse trees, which are used for comparing the similarity between the two trees. The approach of Bunescu and Mooney (2005b) prunes down the feature space off by introducing three sub-kernels. Instead of comparing all possible subsequences of two sentences to count the common ones, their approaches comprises only three types of subsequences: 1) terms before and between two mentions, 2) terms between two mentions, and 3) terms between and after two mentions. The maximal length of each subsequence is further restricted to four according to their observation. Zhao and Grishman (2005) designed kernels for different syntactic sources and used them in their RE approach. For example, all neighbor terms in a sentence are combined to new terms. Based on these new terms,

their approach can derive a bigram kernel. Another example is the dependency path kernel, which is used to compare the similarity of two dependency paths connecting two mentions.

Discussion: Supervised learning strategies can extract relations with high accuracy. However, the major drawback of supervised learning techniques is still that a large amount of labeled data is required to train the models. For the RE task, hand labeling is even more tedious compared to the other two tasks, NER and NEL. This is because the annotation process for the RE task includes extracting and disambiguating named entities from text. As a representative example introduced by Kambhatla (2004), the training set, which contains around $300,000$ words and $9,752$ instances of relations, is exploited for training. Using one of these techniques implies that relabeling and retraining of the model becomes necessary, as soon as either the underlying characteristics of the data sources or the relations of interest change significantly.

Distant supervised learning

As introduced in Section 1.3, the result of RE is one of the key aspects of the knowledge base population task. If we think the other way around, it is possible to exploit existing knowledge from KBs for solving the RE problem. Therefore, one can build up training sets by labeling documents according to the relations contained in KBs. The automatically generated training data can be used to train a supervised learning approach for the RE task.

Snow et al. (2005) collected the hypernym (is-a) pairs from WordNet and annotated sentences with the pairs. They trained multiple classifiers on the annotated dataset, such as multinomial Naive Bayes, complement Naive Bayes, and logistic regression. With the trained model, their approach uses the classifiers to determine if two entities have a hypernym relation.

Mintz et al. (2009) introduced their distant supervision approach using relations contained in Freebase. They build up a training set by labeling Wikipedia articles with Freebase relations. A multi-class logistic classifier is then trained with features such as contextual terms, named entity types, and the dependency path between two entities. In the system DeepDive [Zhang (2015)], the distant-supervised learning method was also applied for extracting relation. Based on the relations contained in Freebase, DeepDive can extract over 20 different types of relations.

Zeng et al. (2014) introduced an RE approach by using CNN. Their approach does not rely on the results of other NLP systems. Zeng et al. (2015) followed a similar idea to acquire weakly labeled training data based on the knowledge in Freebase. Based on the labeled data, piecewise convolutional neural networks are trained for the RE task. This approach applies a multi-instance learning strategy to alleviate the problem caused

by wrong labels in the automatic annotated training set.

Discussion: Distant-supervised learning avoids the hand-labeling process and can build up large-scale training sets efficiently. However, one of the limitations is that it highly relies on the existing knowledge bases, and only the targeted relations must be included. Moreover, the automatic labeling process is often based on the following assumption: If two entities participated in a relation, the sentences that contain both entities should describe that relation. However, this assumption is not always true. Thus, the generated training sets typically contain some wrong labels that can influence the final relation extraction results.

Semi-supervised learning

Semi-supervised learning is another way to avoid the need of hand-labeling training data. Hearst (1992) applied bootstrapping strategy, the approach can extract hypernym relations with given several lexico-syntactic patterns. With an example pattern "*such NP as NP,* (or—and) NP*", the approach can find out that Herrick, Goldsmith, and Shakespeare are authors from the sentence "*... works by such authors as Herrick, Goldsmith, and Shakespeare ...*". The new instances are used to extract new patterns iteratively.

In 1999, Brin (1999) introduced DIPRE, which focused on extracting relations between authors and their corresponding book titles. The system learns patterns based on a set of initial (author, title) pairs (i.e., relation instances) to extract new patterns. To do so, DIPRE expands the seed set with new pairs and repeats the process until the result satisfies a predefined condition.

Agichtein and Gravano (2000) derived the Snowball system from the idea of DIPRE and extended it by introducing a vector-based similarity function to group patterns instead of matching them. Also, some other approaches were developed based on this bootstrapping strategy to extract relations between entities, such as KnowItAll [Etzioni et al. (2004)], StatSnowball [Zhu et al. (2009)], and CPL [Carlson et al. (2010)].

Applying semi-supervised learning strategies can efficiently extract large-scale relations from a small set of initial seeds. In an experiment that was introduced by Pasca et al. (2006), starting from 10 seed instances, the approach can extract 100,000 "Person-BornIn-Year" relations in the first iteration and over 1 million after the second iteration from 100 million Web documents.

Discussion: With the bootstrapping strategy, one can efficiently extract relation from large corpus with only a few hand-craft inputs. A significant problem of semi-supervised learning methods is the semantic drift problem. It happens when some of the seeds that

represent an undesired relation type are included in a seed set. The following iterations can be negatively influenced such that they yield more and more irrelevant seed so that the quality of the final result decreases.

Unsupervised learning

The unsupervised learning approaches typically formulate RE as a clustering task. Without requiring any human inputs, i.e., predefined relations and corresponding labeled data (e.g., training data and seeds), unsupervised learning approaches are aimed at clustering entity pairs according to relations between them.

Banko et al. (2007) defined such a task as *Open Information Extraction Open IE*. In this paper, they introduced their system TextRunner to extract all possible relations from a large corpus. TextRunner can easily adapt to different datasets and extract various relations from them since no hand-labeled inputs are required. In the experiments, TextRunner extracts 60.5 million tuples from 9 million Web pages.

Wu and Weld (2010) proposed the WOE system, which enhances TextRunner by including additional information from Wikipedia articles to construct a training dataset. They use existing knowledge contained in Wikipedia infoboxes to find sentences that describe the specific relations from the corresponding articles. Moro and Navigli (2013) further integrated syntactic and semantic features into an Open IE paradigm. They clustered relational phrases with a kernel similarity measure and used the information to disambiguate extracted relation instances.

Discussion: The Open IE approaches automatically extract all possible relations from a given corpus. However, the semantic information of the results is missing. Since no predefined relation is given, the outputs are only clusters of entity pairs that participate in similar types of relation. The actual meaning of each cluster needs to be recognized through a further process. For example, a typical Open IE system gives only a set of entity pairs in a certain phase that describes their relations. In the case, the relation is described by the phrase "*is author of*". Since there is no definition the type of relation (i.e., author_of relation) this phrase indicates, the system cannot tell that the first mention refers a person who writes a book to which the second mention refers. Therefore, additional processes are required to make the result applicable for further usages.

2.4.3 Datasets

The task of relation extraction was included in MUC-7 [Chinchor and Marsh (1998)], which is introduced in Section 2.2. In MUC-7, as a subtask of the information extraction task, the template relation task is aimed at extracting three types of relations, which

are employee_of, location_of, and product_of. The given dataset consists 200 articles that are evenly split into training and testing subsets.

Table 2.4: List of relation types and subtypes used in ACE 2003 [Kambhatla (2004)]

Type	Subtype	Count	Type	Subtype	Count
ROLE	affiliate-partner	219	**SOCIAL**	associate	119
	citizen-of	450		grandparent	10
	client	159		other-personal	108
	founder	37		other-professional	415
	general-staff	1507		other-relative	86
	management	1559		parent	149
	member	1404		sibling	23
	other	174		spouse	89
	owner	274	**NEAR**	relative-location	288
PART	other	6	**AT**	based-In	496
	part-of	1178		located	2879
	subsidiary	366		residence	395

In the ACE program, one of the tasks is *relation detection and characterization (RDC)*. Several annotated datasets were published for this program. These datasets are widely employed for training and testing RE approaches. In 2002, ACE included the RDC task for the first time [Mitchell et al. (2002)]. The task can be considered as a pilot data consisting of 422 documents. Five types of entities are included (i.e., person, organization, facility, location, and geopolitical entity), which participate in 5 main types (24 subtypes) of relations, including ROLE, PART, AT, NEAR, and SOCIAL. In the following year, Strassel et al. (2003) introduced the ACE 2003 dataset, which includes annotations of relations in Chinese. Kambhatla (2004) presented the detailed statistic of this dataset, which is shown in Table 2.4. In ACE 2004, the overall annotated data in training set increased from 100,000 words to 300,000 words compared to ACE 2003 [Doddington et al. (2004)].

For the RE task, due to the diversity of relation types, approaches are often designed for extracting some specific types of relations. Therefore, similar to NEL approaches, various annotated datasets are created and exploited for testing the performances of different approaches. In Section 4.6, we compared the performance of our RE approach with the distant-supervised approach Zeng et al. (2015) on extracting ownership_of relations between companies from news articles.

2.4.4 N-ary relation extraction

As mentioned earlier, the major focus of the current RE task is to extract binary relations. Beyond binary relations, extracting n-ary relations can also be useful for further

usages. For example, it would be valuable to know the ternary relation of a persons job title at a particular company.

McDonald et al. (2005) introduced a simple solution for this type of problem. They decomposed the n-ary RE task into multiple binary RE tasks. Afterwards, they constructed a graph based on the extracted binary relations, in which the mentions of named entities were treated as nodes that are connected according to the extracted relations. The probability value of each relation that is assigned by the binary classifier is used for selected maximal cliques in the graph.

The dependency information that can be derived from dependency tree parser is useful for solving the n-ary RE problem. Akbik and Löser (2012) introduced their n-ary Open RE approach, KRAKEN. Using the dependency parse (i.e., generated by Stanford Dependencies [10]), KRAKEN extracts n-ary relations from a whole sentence. For example, the extracted relation from the snippet "*... Doublethink, a word that was coined by Orwell in the novel 1984 ...*" is "*Doublethink (was coined by) Orwell (in) 1984*". As an Open IE approach, KRAKEN does not output the semantic meaning of the extracted relations. Recently, Peng et al. (2017) applied Garph *long short-term memory networks (LSTMs)* to extract n-ary relations.

2.4.5 Conclusion

As one of the most valuable information hidden in texts, many further applications can be built based on the result of RE. Due to the complexity of how humans describe relations, it is also time-consuming for humans to identify them from large amounts of text, and knowledge of certain domains is sometimes required. Most of the current approaches make use of contextual information to identify relations between named entities. One problem of solving the RE task is the difficulty of manually creating training sets for supervised learning approaches. Except for the challenges of hand-labeling relations, extra information about mentions of named entities is also required, meaning that one needs to annotate named entities in advance. Therefore, many approaches try to avoid the requirement of a large amount of training data by applying weak-supervised or unsupervised learning strategies. Starting from basic or general relation types, such as hypernym relation, the birthplace of a person, and the capital city of a country, researchers are now focusing more on extracting broader types of relations or some specific complex relations. Furthermore, beyond the binary relations, extracting an n-ary relation is also considered.

[10] https://nlp.stanford.edu/software/stanford-dependencies.shtml

2.5 Our work

In this chapter, we have introduced the related work in the area of named entity mining from textual data from three aspects, named entity recognition, named entity linking and relation extraction. These three tasks can build up an entire pipeline. Mentions of named entities are first discovered and recognized from texts. Then, one needs to disambiguate mentions and link them to corresponding named entities in knowledge bases. In the last step, relations between named entities are extracted based on the information delivered in contexts. In the following two chapters, we present two approaches that separately deal with the named entity linking problem and the relation extraction problem.

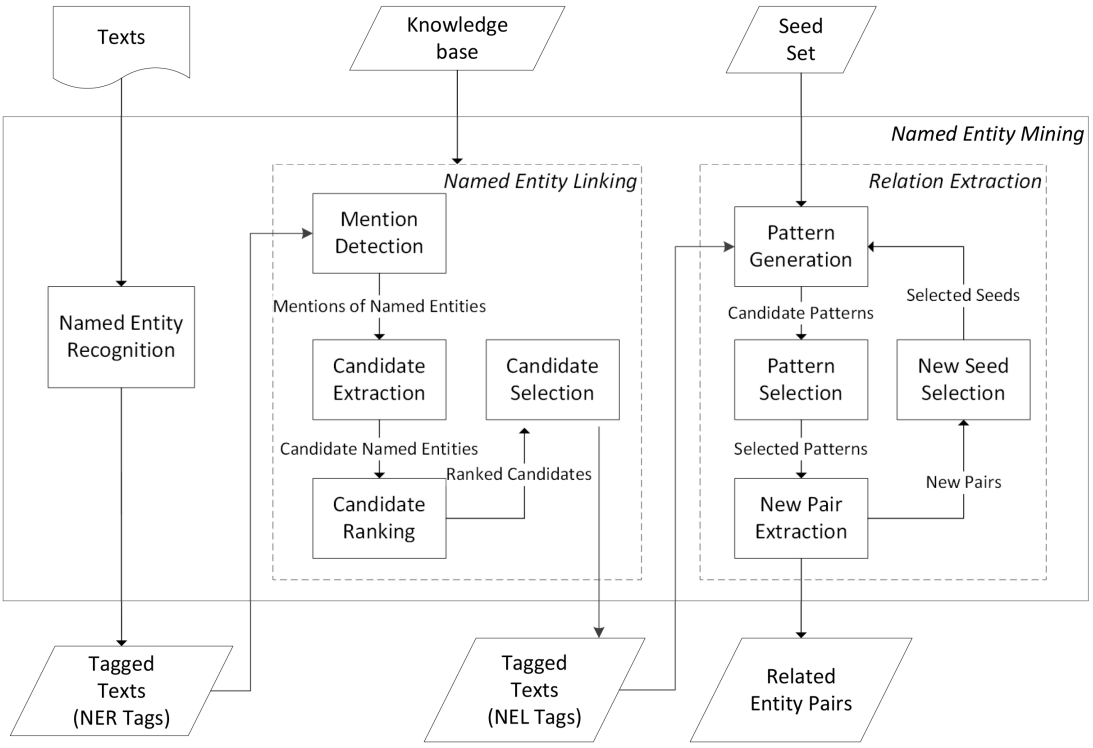


Figure 2.2: Architecture of our named entity mining system

Figure 2.2 illustrates the pipeline of our named entity mining system from NER to RE in detail. In a system, by giving some textual documents, we can extract some particular structural information about mentioned named entities from the documents, including types of named entities (i.e., NER tags), particular named entities to which mentions refer(i.e., NEL tags), and relations between named entities (i.e., related entity pairs). First, due to the long history of the focuses on the NER task, some well-developed systems are available online, such as the Stanford NER Tagger [Finkel et al. (2005)], and Apache OpenNLP[11]. Therefore, we can apply existing NER approaches in the next steps.

[11] https://opennlp.apache.org/

Second, we developed a light-weighted named entity linking approach called *Bagging for Entity Linking (BEL)* [Zuo et al. (2014)], which employs the Stanford NER Tagger to recognize mentions of named entities. For each extracted mention, BEL tries to identify the correct named entity in a KB that to which a mention refers (see Chapter 3). Finally, for the NEL task, we developed a semi-supervised approach and focused on extracting business relations between companies [Zuo et al. (2017)]. For our RE approach, we simplified BEL to disambiguate company mentions as a preprocessing step. Our approach can efficiently extract target business relations with only little hand-labeled data (see Chapter 4).

CHAPTER 3

Named Entity Linking

In this chapter, we introduce a novel named entity linking approach, *bagging for entity linking* (BEL). With recent advances in the areas of knowledge engineering and information extraction, the task of linking textual mentions of named entities to corresponding entity IDs in a knowledge base has received much attention. The rich, structured information in state-of-the-art knowledge bases can be leveraged to facilitate the difficult task of disambiguation, i.e., linking an ambiguous mention to the correct entity ID. Although recent approaches achieve satisfactory accuracy results, they typically suffer from at least one of the following issues: (1) the linking quality is highly sensitive to the amount of textual information; typically, long textual fragments (e.g., whole paragraphs or documents) need to be processed in order to capture the context of a mention and to reliably disambiguate it, (2) the disambiguation uncertainty is not explicitly addressed and often only implicitly represented in the ranking of entities to which a mention could be linked, (3) complex, joint reasoning negatively affects the disambiguation efficiency. BEL addresses all the above issues by (1) operating on sliding windows, (2) adequately aggregating decisions from an ensemble of simple classifiers, (3) following a local reasoning strategy by exploiting previous decisions whenever possible. In an extensive experimental evaluation on hand-labeled and benchmark datasets, our approach outperformed state-of-the-art entity linking techniques, both in terms of quality and efficiency. BEL employs an ensemble-based disambiguation approach that exploits the terms that surround a textual mention to best capture its context, and a parsimonious linking model to combine the above method with a prior probability. We apply the language model [Zhai and Lafferty (2004)] capture the similarity between the context of a mention and the terms describing possible candidate entities in Wikipedia.

The remainder of this chapter is organized as follows: We introduce a high-level overview of our named entity linking approach in Section 3.2. Section 3.3 represents the core component of our BEL approach, named entity disambiguation. In Section 3.4 we introduce how to deal with the entities that are not included in knowledge bases. The

experimental evaluation is presented in Section 3.5. The detailed introduction of the related work can be found in Section 2.3. This chapter is based on the work [Zuo et al. (2014)] and the related work is introduced in Chapter 3.

3.1 Named entity linking with knowledge bases

As we have introduced in Section 1.2, NEL is challenge task due to the ambiguity of textual mentions. Resolving these ambiguities is often referred to as NED. In the latter setting, the proliferation of clean knowledge bases with rich semantic relations between Web entities, e.g., DBpedia, Freebase, or YAGO, has given rise to novel, reliable NEL techniques that exploit the semantic relatedness between entities for the linking process [Hoffart et al. (2011b, 2012); Shen et al. (2012)]. The approaches presented in [Hoffart et al. (2012)] and [Hoffart et al. (2011b)] use this kind of relatedness in a collective reasoning process, where multiple mentions are jointly mapped to named entities in the knowledge base. A score measuring the *coherence* of this joint mapping, and two other scores – one concerning the contextual similarities between mentions and entities and another one representing the prior probability that an entity is referred to by a mention – are linearly combined in an objective maximization function. While each of the mentioned scoring components plays an important role in the linking process, the coherence score has been identified as the key to ensure a reliable linking [Hoffart et al. (2011b)].

However, the complex, joint reasoning extremely decreases the efficiency of NEL approaches, so that those approaches cannot handle a large amount of text documents. Our lightweight disambiguation model builds on a majority-voting strategy that employs a bagging of multiple ranking classifiers, thus the name BEL: Bagging for Entity Linking. Each ranking classifier operates on a randomly sampled subset of terms surrounding the mention in focus. These terms are sampled from a so-called textual *range of relevant terms*, i.e., terms that are most promising for determining the context of the mention. Finally, based on the sampled terms, each ranking classifier proposes a ranked list of candidate entities and the mention is linked to the entity that is proposed as top-ranked candidate by the majority of the classifiers.

In summary, the main contributions of this work are:

1. A novel ensemble-based disambiguation approach that exploits the terms that surround a textual mention to best capture its context; a parsimonious linking model that combines the above method with a prior probability (similar to the one presented in [Fader et al. (2009)], [Hoffart et al. (2011b)], or [Lin et al. (2012)]) of a candidate named entity being referred to by a given mention yields a highly efficient linking process.

2. An analysis of the disambiguation impact of the components used in BEL on the

final linking decision.

3. A detailed quality and efficiency comparison with the state-of-the-art methods of [Cucerzan (2007)], [Hoffart et al. (2011b)], and [Hoffart et al. (2012)] on multiple real-world and synthetic datasets; apart from being more efficient, BEL also achieves a linking quality that is comparable to or even better than that of the above methods.

3.2 High-level overview of BEL

In this section, we introduce the a high-level overview of the BEL approach. The processing pipeline can be found in Figure 2.2. The major assumption we make is that the textual corpus from which the selected knowledge base has been derived is freely available. For example, the textual corpus of knowledge bases such as YAGO or DBpedia is Wikipedia, which is an open source of information about the entities in the two knowledge bases.

In this work, the focus is not on the recognition of named entity mentions in a text but rather on the disambiguation process once the mentions are known. Throughout this work, we assume that a reliable named entity recognition tool is available. The Stanford Parser [de Marneffe et al. (2006)] is such a tool; the BEL approach relies on it to recognize textual mentions of named entities. Once the mentions have been recognized, BEL retrieves promising candidate entities from the knowledge base and employs a sophisticated language-model-based algorithm to best exploit the textual context of the mentions for the disambiguation process. The method is described in the following subsections.

Exemplarily, we use the YAGO knowledge base to highlight the main idea of the algorithm, which is shown in Algorithm 1. YAGO is a knowledge base with structured information about a large proportion of the entities contained in Wikipedia, and thus a popular representative of many knowledge bases derived from Wikipedia.

Once the set of mentions has been derived from a given document (line 1), for each mention, a list of promising candidates is derived from Wikipedia. The candidates are ranked by a so-called "prominence" score, which is computed as the probability of a Wikipedia article (i.e., the entity represented by the article) being referred to by the mention (lines 2, 3). This probability can be inferred from the intra-Wikipedia links and the Wikipedia Redirect Pages (see Section 3.3.1). In case the list of candidates is empty, the corresponding mention is linked to a designated entity, E_{NULL}, which represents entities that are not in the knowledge base (lines 4, 5). The same holds for the case that the top-ranked candidate occurs in Wikipedia but not in YAGO (lines 7 - 9). Otherwise, the joint majority decision of multiple ranking classifiers is computed as follows: Each ranking classifier samples terms with replacement from a sliding window that represents

the range of terms that are relevant with respect to the context of the mention. The sampled terms are used to compute the contextual similarity between the mention and candidate Wikipedia articles. Each ranking classifier combines this contextual similarity score with the above "prominence" score (which describes how probable a candidate entity is for a given mention). The combined score yields the final ranking (lines 11 - 13) of each classifier. If the majority of the ranking classifiers has the same candidate as a top-ranked entity, the mention is linked to that candidate. Otherwise, the candidate is linked to E_{NULL} (lines 14 - 18).

Algorithm 1 Bagging for Entity Linking Algorithm

Input: document file $\mathbf{D} = (t_1, t_2, ...)$, HashMap $\mathbf{V}$ that maps the ID of a ranking classifier to the top-ranked candidate entity by that classifier
Output: linkage between mentions $\mathbf{M} = \{m_1, m_2, ...\}$ in $\mathbf{D}$ and corresponding entities $\mathbf{E} = \{e_1, e_2, ...\}$ in YAGO

1: $\mathbf{M}$:= *recognizeMentions*($\mathbf{D}$)
2: **for** each mention $m_i \in \mathbf{M}$ **do**
3: $\quad\mathbf{L}_{m_i}$:= *getTopKCandidates*(k,m_i) /*according to the "prominence" score $S_{PR}(e, m_i)$*/
4: $\quad$**if** $\mathbf{L}_{m_i}$ is empty **then**
5: $\quad\quad$link m_i to E_{NULL} /*i.e.,mention cannot be linked*/
6: $\quad$**else**
7: $\quad\quad e' := \text{argmax}_{e \in L_{m_i}} S_{PR}(e, m_i)$
8: $\quad\quad$**if** e' is not in YAGO **then**
9: $\quad\quad\quad$link m_i to non-YAGO entity E_{NULL}
10: $\quad\quad$**else**
11: $\quad\quad\quad$**for** each ranking classifier S_n **do**
12: $\quad\quad\quad\quad\mathbf{V}$.put($n$, argmax$_e$($Sim(e, S_n, m)$))
13: $\quad\quad\quad$**end for**
14: $\quad\quad\quad$**if** an e^* occurs more than $\frac{|\mathbf{V}|}{2}$ in $\mathbf{V}$.*values*() **then**
15: $\quad\quad\quad\quad$link m_i to e^*
16: $\quad\quad\quad$**else**
17: $\quad\quad\quad\quad$link m_i to non-YAGO entity E_{NULL}
18: $\quad\quad\quad$**end if**
19: $\quad\quad$**end if**
20: $\quad$**end if**
21: **end for**

In addition, for efficiency reasons, BEL exploits previous disambiguation decisions whenever possible. If a mention occurred multiple times in a document and was already reliably linked b times to the same named entity in YAGO, the previous linking decisions (for that mention) are reused without rerunning the disambiguation process. This heuristic may lead to false positives, e.g., London may be incorrectly linked to Jack London, but in empirical evaluations on real-world datasets, the algorithm has shown a robust quality behavior while being highly efficient (see Section 3.5).

The runtime of the above algorithm is dominated by the computation of the contextual similarity scores of each ranking classifier. More specifically, each classifier needs

$O(NlogN)$ steps to propose a context-based ranking of the N candidates derived by the "prominence" score. Since the classifiers operate independently from each other, the algorithm allows parallel computation of the contextual similarity scores. However, in this work, we have implemented a sequential version, which for K different ranking classifiers has a complexity of $O(KNlogN)$.

3.3 Disambiguation process

For the reliable linking of a mention to an entity in the knowledge base, it is crucial to have a candidate ranking strategy that quickly prunes the candidate space by considering only the most probable entities for a given mention and its context. More specifically, let $S_i(m)$ denote the textual context of a mention m. Given $S_i(m)$ and m, the goal is to find an entity e from the knowledge base that maximizes the probability $P(e|S_i(m), m)$. This probability can be reformulated as follows:

$$\begin{aligned} P(e|S_i(m), m) &= \frac{P(S_i(m)|m, e)P(m|e)P(e)}{P(S_i(m)|m)P(m)} \\ &= \frac{P(e|m)P(S_i(m)|m, e)}{P(S_i(m)|m)} \\ &\propto P(e|m)P(S_i(m)|m, e) \end{aligned} \tag{3.1}$$

Note that $P(S_i(m)|m)$ is a constant for different entities in the same candidate list as it is independent of entities. Hence, $P(S_i(m)|m)$ can be omitted without influencing the ranking of entities in a candidate list.

In practice, we estimate the above probability based on two criteria. First, $P(e|m)$ is estimated by the "prominence" score $S_{PR}(e, m)$, which is based on the frequency of the entity e being referred to by the mention m. We calculate this score based on the information contained in Wikipedia. Second, we estimate $P(S_i(m)|m, e)$ as:

$$\begin{aligned} P(S_i(m)|m, e) &\approx P(S_i(m)|e) \\ &\propto S_{CS}(S_i(m), e) \end{aligned} \tag{3.2}$$

where a contextual similarity score $S_{CS}(S_i(m), e)$ is employed to estimate the $P(S_i(m)|m, e)$. Note that $S_i(m)$ typically depends on the entity e and not on the mention m. Combining the above two components, we can rank the candidate entities by a similarity score

$Sim(e, S_i(m), m)$, which is an estimation of $P(e|S_i(m), m)$ as follows:

$$\begin{aligned} Sim(e, S_i(m), m) &= \log P(e|S_i(m), m) \\ &\propto \log P(e|m) + \log P(S_i(m)|m, e) \\ &\propto S_{PR}(e, m) + S_{CS}(S_i(m), e) \end{aligned} \tag{3.3}$$

This estimation also follows the intuition about human behavior when trying to disambiguate a named entity. For instance, consider the following text: "*Jordan is an American former professional basketball player in the National Basketball Association (NBA), who has won the Most Valuable Player (MVP) Award.*" Although this example poses a simple disambiguation task for us humans, the interesting question is how we actually disambiguate the mention "Jordan". Definitely, the contextual information gives us a strong hint that "Jordan" refers to a basketball player, but not to the country of *Jordan* or the FormulaOne team *Jordan Grand Prix*, etc. However, there are multiple basketball players by the name "Jordan" in the NBA. Nevertheless, most of us would immediately assume that "Jordan" refers to *Michael Jordan*, who is the most well-known basketball player with that name. Therefore, we think that the above two components $S_{PR}(e, m)$ (see Section 3.3.1) and $S_{CS}(S_i(m), e)$ (see Section 3.3.2) should build the basis of any named entity disambiguation system. BEL builds on such a model to solve the named entity linking task.

We first explain the strategy for retrieving the list of candidate entities that are frequently referred to by a given mention and then introduce the computation of contextual similarity.

3.3.1 Candidate Ranking by prominence scores

In general, to estimate the probability of an entity being referred to by a mention m, a large natural-language text corpus would be needed, in which all entities that may occur with m are known and all occurrences of m are labeled with the corresponding entity. For many valuable named entities on the Web, Wikipedia represents such a corpus.

For knowledge bases, such as YAGO or DBpedia that have been derived from Wikipedia, we can leverage the hyperlinked Wikipedia entities to infer the "prominence" of an entity for a given mention. A useful notion in this context is that of a *surface*. A surface is a textual mention that is hyperlinked to a Wikipedia article that describes a named entity. For instance, *Michael Jordan* could occur in a text with the surface "Michael Jordan", "Jordan", "Air Jordan", and so on. In order to derive candidate entities for a given surface, we compute the relative frequency by which a Wikipedia article is hyperlinked from any occurrence of this surface. Note that the surface can also occur as part of a redirect link. Hence, for a given surface m, we compute a distribution $P(e|m)$ on all

named entities e (i.e., Wikipedia articles) that are referenced from m. The candidate entities for m are ranked by $P(e|m)$. There may be hundreds or even thousands of candidates in the ranked list, and finding the right named entity among thousands of candidates can be highly challenging and inefficient. However, we have noticed that, when ranking by $P(e|m)$, the correct entity is typically contained among the top-40 candidates (see Figure 3.3 on page 65), since the entities in the tail of a candidate list do not frequently occur with the corresponding surface. Moreover, the coverage rate, i.e., relative frequency by which the correct entity is contained in the top-k candidates, increases notably for k between 1 and 40 and stabilizes quickly with k growing beyond 40. In order to remain efficient and still account for difficult disambiguation cases (i.e., for which the correct candidate may occur even lower in the list) we have empirically established a threshold of $k = 40$. As it can be seen in Figure 3.3, in the overwhelming majority of the cases, the correct entity will be among the top-40 candidates. Another hypothesis is applied to prune the candidate list, when the most prominent candidate is not in YAGO, the mention is considered to be a non-YAGO entity. Note that such a pruning step is essential for an efficient named entity linking strategy.

3.3.2 Bagged language models for contextual similarity

It is important to note that the above candidate selection strategy is based only on the "prominence" of entities with respect to mentions, and the contextual information is not taken into account yet. For the previous example of "Michael Jordan", the corresponding named entity is only in the 15th position of the candidate list which is ranked by the "prominence" score, while the country "Jordan" is in the first position. However, by exploiting the contextual information, BEL can identify that "Jordan" should be mapped to the famous basketball player, *Michael Jordan*.

For any NEL approach, exploiting the contextual information in the best possible way in order to enable a correct disambiguation is a challenging task. Our approach, BEL, has to identify the correct entity among the top-40 candidates based on contextual clues. To this end, BEL builds on a bagging of multiple statistical language models, each of which assigns a contextual likelihood score to every candidate entity in the list.

First of all, let m be a mention occurring in the text. Further, we assume that the semantic relatedness between a term and a mention depends on their distance in the text (in terms of words between them). When a term is closer to a mention, the relatedness between them is higher. We evaluate this assumption in the evaluation section and find that typically a range of 55 consecutive terms surrounding the mention (i.e., the mention typically occurs in the center of the range[1]) is enough for our algorithm to optimally capture the context of the mention. Such a textual range is typically much smaller than the whole text of the document, and is implemented as a sliding window over the document. The local and independent reasoning strategy based on sliding windows can

[1] Except for very short documents, where the mention can occur in any position.

also improve the efficiency performance of BEL. We denote the set of terms in such a range by $R(m)$. Its contextual similarity to a candidate entity e can be quantified by means of a statistical language model of e.

Statistical language models [Zhai and Lafferty (2004)] have been widely used in Information Retrieval for ranking result documents to keyword queries. It is a probability distribution over terms. A document $d = (t_1, ..., t_m)$ can be represented by a language model $L(\boldsymbol{\theta}_d)$. By given $L(\boldsymbol{\theta}_d)$, one can calculate the likelihood $P(q|L(\boldsymbol{\theta}_d))$ that a query q is a sample from the document d. In our case, we employ a language model for capturing the contextual similarity between a candidate entity e and a mention m for the given relevant range $R(m)$, which can be defined as follows:

$$P(R(m)|L(\boldsymbol{\theta}_e)) \approx \prod_{t \in R(m)} [(1-\lambda)P(t|L(\boldsymbol{\theta}_e)) + \lambda P(t)] \tag{3.4}$$

where $L(\boldsymbol{\theta}_e)$ represents a probability distribution (i.e., multinomial) over terms occurring in the context of e (which we will define subsequently), $P(t)$ is a prior probability of the term t, (i.e., computed as the relative frequency of t in the Wikipedia corpus), and λ is used as a smoothing parameter, to account for terms t for which $P(t|L(\boldsymbol{\theta}_e)) = 0$ but $P(t) > 0$ [Zhai and Lafferty (2004)]. In our experiments, we set the value of λ to be 0.5 as a default value to fit general cases. Figure 3.1 gives a schematic overview of the ingredients for the computation of contextual similarity scores based on a statistical language model.

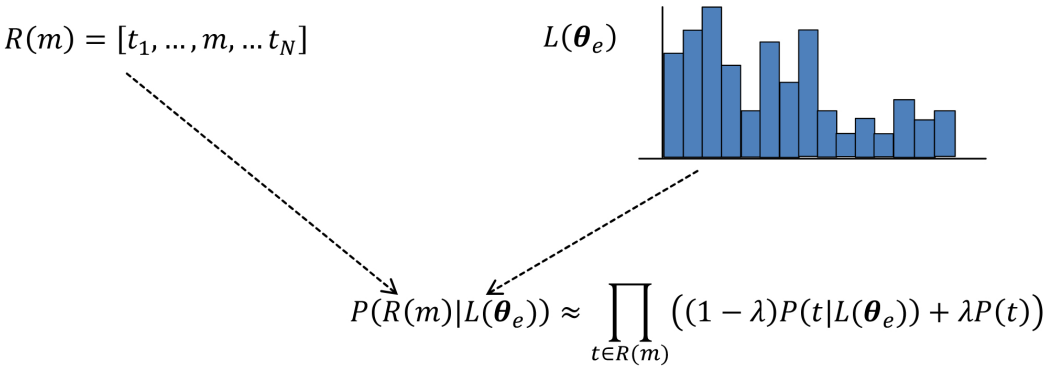


Figure 3.1: Application of a statistical language model to the computation of the contextual similarity score between a range of relevant terms $R(m)$ for a mention m and a candidate entity e represented by its language model.

In order to generate the language models of all Wikipedia entities, we indexed the corresponding Wikipedia articles and, after removing stopwords and words with non-English

characters, we derived frequencies for each term occurring in the articles. Furthermore, since in our experiments we have used the YAGO knowledge base, from the articles corresponding to neighbors of a YAGO entity e (i.e., YAGO entities that stand in a YAGO relationship to e), we derived terms that frequently occur in the context of e (i.e., terms that occur close to e in the article or have a hyperlink to the article of e). For every entity e, all term frequencies were normalized, thus yielding a term distribution, from which we derived the language model $L(\boldsymbol{\theta}_e)$. Note that $\boldsymbol{\theta}_e$ is the vector containing all the parameters of the above distribution (i.e., the maximum likelihood estimations for the probability of a term given the context of e).

The contextual similarity score is computed as follows:

$$S_{CS}(R(m), e) = \frac{\log P(R(m)|L(\boldsymbol{\theta}_e))}{|R(m)|}. \tag{3.5}$$

Note that from an information-theoretic viewpoint the negation of this score corresponds to number of bits needed to encode a term from $R(m)$ on average when $L(\boldsymbol{\theta}_e)$ is known; the better the terms in $R(m)$ fit the term distribution in $L(\boldsymbol{\theta}_e)$ the fewer bits are needed.

Note that in Equation 3.4, according to the independence assumption for the term occurrence probabilities given the language model, when the length of the relevant range grows, more and more correlations between terms will be dismissed. In order to mitigate the impact of this independence assumption on terms in a long sliding window, and to capture a high contextual diversity for the mention in question, we rely on the principle of bootstrapping, i.e., subspace sampling: This means that for a mention m, we generate sets of words by randomly drawing terms from $R(m)$ based on a specific sampling strategy (i.e., bootstrapping) [Breiman (1996)]. In its most basic form, bootstrapping consists of b uniformly sampled elements, with replacement, from a training set of b elements. Computing the probability of an element not being selected after the b samples leads to $(1 - \frac{1}{b})^b$ and can be approximated by $\exp(-1)$ for large b. Hence, in general, this sampling procedure results in multiple subsets that, on average, are smaller than the original training set (i.e., containing approximately $\exp(-1)$=63.2% of the elements in the original set). For our setting, this means that a generated subset contains on average approximately 63.2% of the terms from the range $R(m)$, thus mitigating the influence of the independence assumption on the large range. However, the main advantage of this bootstrapping procedure is that it increases the contextual diversity of the mention by enabling different representations of its context. For each such representation (i.e., for each subset $S_n(m)$) we employ a language model classifier v_n that ranks the candidate entities by contextual similarity to $S_n(m)$ according to Equation (2). Finally, each classifier v_n takes also the "prominence" score of the candidates into account to compute the final ranking by $Sim(e, S_n(m), m)$. An illustration of the above bagging strategy is given in Figure 3.2. The effect of employing bootstrapping is empirically evaluated in Section 3.5.5.

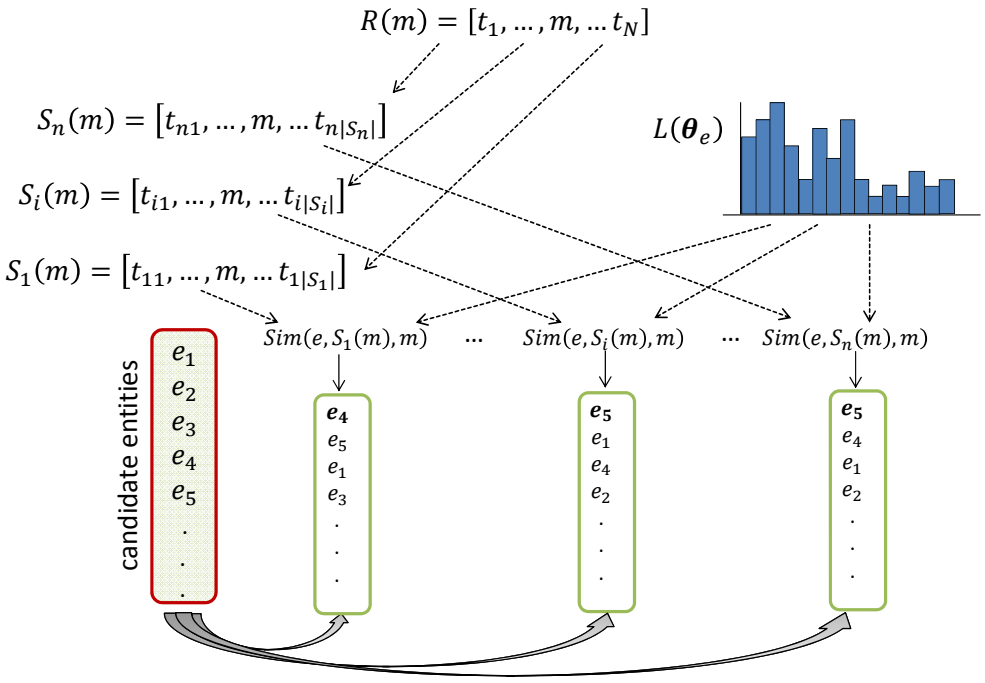


Figure 3.2: Strategy for generating a bagging of language models, each of which operates on a randomly sampled subset $S_n(m)$ of the original relevant range $R(m)$ and assigns contextual similarity scores to the candidate entities based on that subset.

Note that this pluralistic scoring strategy, based on multiple subsets with diverse contextual information, gives a better understanding of the confidence in the disambiguation process. The final majority voting process is as follows:

Let $\mathbf{V}$ be the vector that contains the top candidate entity proposed by the n'th classifier in its n'th component (i.e., $e_n = \text{argmax}_e(Sim(e, S_n(m), m))$). Let e^* be the entity that occurs most frequently as a component of $\mathbf{V}$. If e^* occurs more than $|\mathbf{V}|/2$ times in $\mathbf{V}$, the mention m is linked to e^*, otherwise it is linked to E_{NULL}, which means the referred entity is not in the knowledge base.

3.4 Recognizing non-YAGO entities

For an improved accuracy of the linking process, it is also crucial to reliably recognize true negatives, i.e., mentions that correspond to entities that are not present in the underlying knowledge base. In this work, we define the named entities that are not included in YAGO as non-YAGO entities. We applied various strategies to classify

whether a mention of named entity is not in knowledge base.

First, when the retrieved list of candidates is empty, the mention is classified as a non-YAGO entity. In addition, in case of the YAGO knowledge base, we check whether the most prominent Wikipedia entity for a given mention is presented in YAGO; if this is not the case, the mention is classified as a non-YAGO entity. Note that many entities from Wikipedia are not present in YAGO, either due to recently added articles or to articles that represent concepts; in YAGO, the concepts have been derived from WordNet.

Second, a flexible threshold is used to recognize a non-YAGO entity. It is calculated as the sum of the default "prominence" score and contextual similarity score. The "prominence" score is the maximal score among the Wikipedia entities in the candidate list that is not present in the knowledge base, or a default "prominence" score when there is no such entity. Both of those two default scores are simply derived from the smoothing of the probability estimations (i.e., Laplace smoothing for the "prominence" score and background smoothing for the contextual similarity score). If none of the candidates has a higher score than the threshold, the corresponding classifier proposes to link to the non-YAGO entity.

Finally, as introduced in the previous section, when no entity in a candidate list has been top-ranked over 50% of the classifiers, the corresponding mention is recognized as a non-YAGO entity.

Although these strategies are relatively straightforward, they lead to a notable improvement in the recognition of true negatives. Further investigation of more elaborate strategies for the reliable detection of true negatives is part of our future work agenda.

3.5 Experimental evaluation

To evaluate the performance of our approach, we applied BEL as well as three other competitor approaches on various datasets. As the experiment results shows, that our approach outperforms the state-of-the-art techniques, both in terms of quality and efficiency.

3.5.1 Datasets

Three different datasets are used for evaluating our named entity linking approach.

CoNLL-YAGO: The CoNLL-YAGO dataset is based on the CoNLL 2003 data [Tjong Kim Sang and De Meulder (2003)], which contains 1,393 Reuters articles[2]. In the AIDA project [Hoffart et al. (2011b)], named entities that were recognized by the Stanford

[2] http://www.reuters.com/researchandstandards/

parser [Finkel et al. (2005)] in the corpus have been manually linked to the corresponding entities in YAGO2 [Hoffart et al. (2011a)]. In a manual inspection of the dataset, we found that it contained links and annotations with which we disagree, for example, "EU" was annotated as a non-YAGO2 entity (i.e., entity that does not occur in YAGO2), whereas we think that a linking to the entity "European Union" (in YAGO2) would have been correct. Another example is, "British" followed by the word "lamb", which has been linked the named entity "United Kingdom" in YAGO2. Considering that the complete mention is actually "British lamb" in the text, we think that it is not a named entity and removed its label. Therefore, we decided to partly relabel 76 randomly picked articles, in order to correct the annotations that we found to be wrong, and used them for the experiments.[3] Please note that this manual check and annotation of the mentions in the articles can be quite tedious and time-consuming, as many ambiguous entities have to be disambiguated and linked to the correct entities in the YAGO2 knowledge base. Note that despite the relatively moderate number of articles that we use for evaluation (i.e., 76 labeled documents out of 1,393), the measured confidence of the evaluation results is quite high; we report the 99% confidence interval (computed on random subsamples of the dataset) for all the approaches and all the evaluation measures used.

CUCERZAN: This dataset consists of 350 Wikipedia articles that were randomly selected by S. Cucerzan to evaluate his approach [Cucerzan (2007)]. All the annotated named entities in this corpus are named entities that occur with hyperlinks in the 350 Wikipedia articles of the corpus. Since this dataset has been created in 2006 (more specifically: 9/11/2006), we had to recover several articles from the old Wikipedia version. However, not all articles could be found, so the evaluation dataset consists of 336 out of the 350 articles of the original corpus. Furthermore, since our competitor AIDA-KORE (see Section 3.5.5) throws exceptions when processing two of the articles, we had to exclude those two articles from the dataset corpus to make the evaluation possible.

KORE: The KORE dataset is the smallest among the datasets that we have used for evaluation. This dataset, as well, has been produced for the AIDA project [Hoffart et al. (2011b)]. It is a synthetic corpus consisting of 50 very short articles, where each article contains one or more hand-crafted sentences about different ambiguous mentions of named entities. An example article is "David and Victoria added spice to their marriage". This dataset is quite difficult, as there is very little context on the numerous entities occurring in the sentences. As Table 3.1 shows, about 24.67% of the words in the article are named entities. Furthermore, the named entities in this corpus are mentioned in a simple format, e.g., "Stanford" may refer to *Stanford University* and "Steve" to *Steve Jobs*, which, together with the sparse context, adds to the difficulty of the automated disambiguation task.

As a knowledge base for evaluating we used YAGO2 [Hoffart et al. (2011a)], which has also been used for the AIDA project [Hoffart et al. (2011b, 2012)].

[3]The annotated dataset can be found at: https://hpi.de/naumann/projects/completed-projects/bel-entity-linking.html

Table 3.1: Datasets overall information

	CoNLL-YAGO	CUCERZAN	KORE
articles	76	336	50
mentions (total)	1431	5343	148
mentions (non-YAGO)	279	936	7
word count (avg.)	173	384	12

3.5.2 Evaluated approaches

We compare BEL to two other prominent approaches, AIDA [Hoffart et al. (2011b, 2012)] and the Cucerzan approach (in the following, referred to as LED) [Cucerzan (2007)], which, as reported in the corresponding papers, are amongst the best-performing approaches and achieve high quality in the disambiguation and linking task. Experience-wise, we can confirm that the very recent AIDA approach has indeed raised the bar for many entity linking methods. Empirically, it shows a highly reliable behavior even with respect to difficult disambiguation tasks.

The AIDA approach comes in different versions: In its original version [Hoffart et al. (2011b)], it provides an algorithm that uses a graph-based connectivity between candidate entities of multiple mentions (i.e., graph coherence, e.g., derived from the *type, subclassOf* edges of the knowledge graph or from the incoming links in Wikipedia articles) to determine the most promising linking of the mentions. We refer to this version of AIDA as AIDA-GRAPH. In another version that has been optimized for datasets such as KORE [Hoffart et al. (2012)], AIDA's coherence model has been extended to recognize key-phrases for named entities, which are then used to determine a similarity score based on key-phrase overlap between candidate entities. We refer to this version as AIDA-KORE.

The Cucerzan approach extends the term-based feature vectors of Wikipedia entities by information from other articles that link to it, but instead of using the whole article text only some key phrases and immediate Wikipedia categories are included. The goal is to find a linking of mentions to Wikipedia entities, such that the sum of vector-based similarities between the candidate entities and the document (containing the mentions) as well as the similarities between pairs of candidate entities is maximized. We refer to this method as LED (for Large-scale Entity Disambiguation). The original work has been conducted at Microsoft and the code is proprietary. Hence, we re-implemented the algorithm according to the descriptions in the paper. We made sure to implement it as well as possible, by evaluating it on the original dataset, i.e., CUCERZAN, and achieving results that are comparable to those presented in the original paper. Note that, however, some of the entities that occur in the CUCERZAN dataset refer to concepts (e.g., summer, time zone, keyboard, etc.). These entities are not contained in YAGO2, since all the concepts in YAGO stem from WordNet [Fellbaum (1998)] and thus have names that are different from those used in Wikipedia. In this sense, the task of link-

ing mentions of the CUCERZAN dataset to YAGO is different from the original task addressed in [Cucerzan (2007)], where mentions were linked to Wikipedia articles.

For our approach, BEL, the parameters (i.e., the threshold on the number of selected candidates and the size of the range of relevant terms) are optimized to deal with common natural-language articles on the Web (e.g., such as those occurring on encyclopedic pages or news sites). The same parameter settings are used for all the three datasets described above to show the performance of BEL on different types of corpora. This is comparable to the AIDA-GRAPH approach, which also uses the same parameters for all the datasets. In contrast, the AIDA-KORE approach has been optimized to handle short texts with sparse context, such as those occurring in the KORE dataset. We discuss the exact parameter settings of BEL in the following subsection.

3.5.3 Common setting of BEL parameters

We can improve the overall performance of the BEL approach by analyzing and appropriately adjusting several key parameters. An optimal setting of the parameters was empirically established, based on all three datasets CoNLL-YAGO, CUCERZAN and KORE. The articles contained in CoNLL-YAGO and CUCERZAN are real-world texts while the articles in KORE are manually generated short texts. Overall, there is a high variability in the textual characteristics of the three datasets. Therefore, the setting derived from these three different datasets can be used as the common setting of BEL for linking named entities in general cases.

Threshold for the selection of candidates

In our approach, each mention is assigned a list of candidates. In general, such a list could contain hundreds or even thousands of entities. However, there is at most one entity in the list to which the corresponding candidate should be linked. On one hand, the more entities in the list, the higher the probability that the correct one is included; on the other hand, longer candidate lists contain more noise, thus rendering the disambiguation task more difficult. We randomly pick 1000 mentions occurring in the three datasets to analyze the impact of the candidate list size. The coverage rate (i.e., the relative frequency by which the correct entity is contained in the list) in relation to the list size is shown in Figure 3.3. Note that the lists are sorted by decreasing "prominence" scores (see Section 3.3.1). In this experiment, 139 out of 1000 mentions have no corresponding entity in the knowledge base, while 61 candidate lists do not contain the correct entity, which means, the maximum coverage rate that our candidate selection strategy can achieve is $800/861 \approx 92.92\%$. As the curve shows, in more than 85% of the cases, the correct entities are indeed located within the top-5 positions of the candidate lists. Furthermore, most of the correct entities are located within the top-40 positions, which shows that, mostly, it is not necessary to keep the whole candidate list for disambiguation. Therefore,

we decide to select the top-40 candidates from each of the candidate lists for further processing. For all versions of the BEL approach this threshold is set to this constant value, which leads to an acceptable efficiency behavior, while still providing certain coverage guarantees.

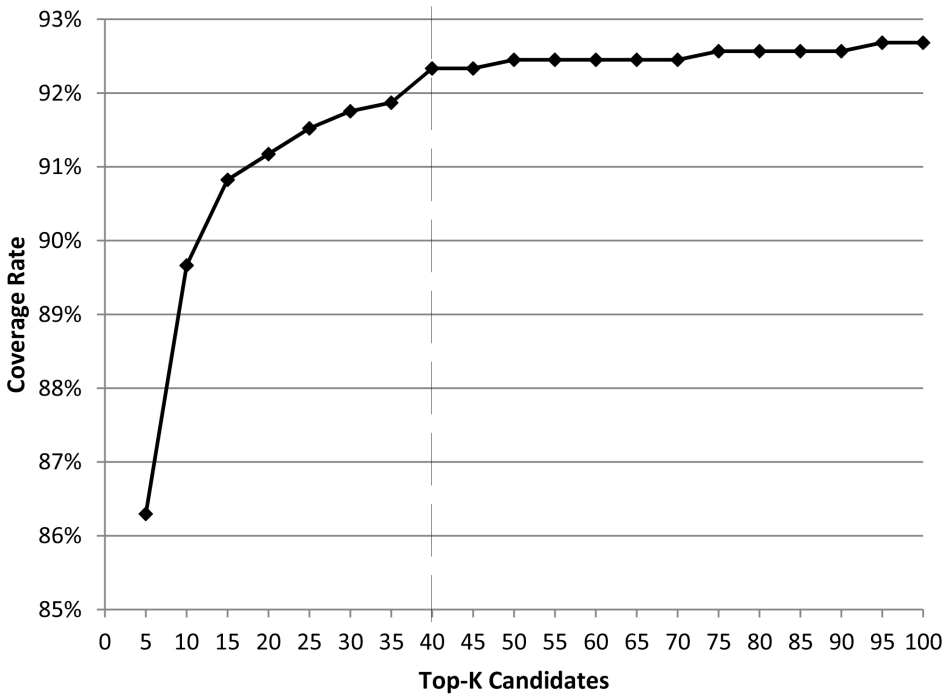


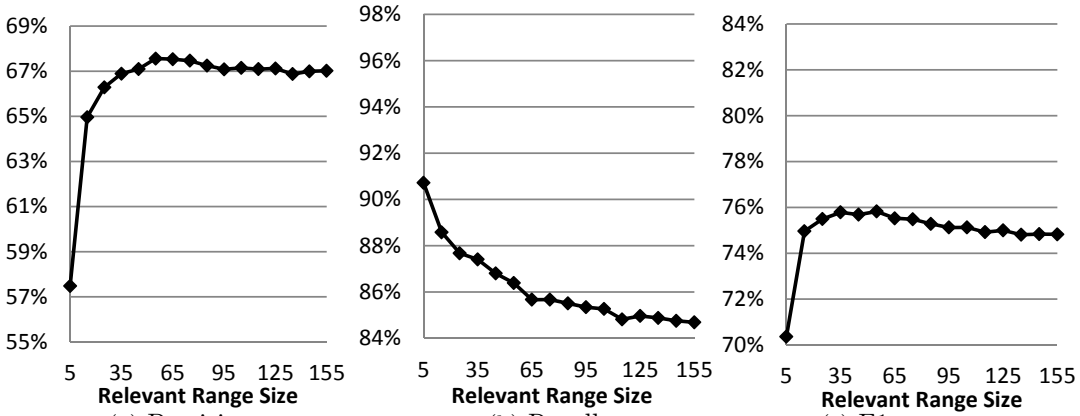
Figure 3.3: Correct Entity Coverage Rate (in %).

Language model parameters

The bagged language models are aimed at capturing the contextual information of a mention by randomly sampling terms surrounding the mention, a process that is repeated several times and leading to several random textual subsets derived from a range of relevant terms. Note that this subspace sampling approach encourages the diversity of contextual information derived from the above range. Moreover, the agreements between the language-model-based rankings for the different subsets can help mitigate the noise present in the whole relevant range. The quality of this bagged language model algorithm depends mainly on two criteria: (1) the size of the range of relevant terms, and (2) the number randomly sampled subsets.

The impact of the size of the relevant range on the linking quality is more intricate, in the sense that the larger the range, the more noise will be included in the contextual information about the mention. On the other hand, the shorter the range, the sparser

the available contextual information. Therefore, finding a good trade-off between noise and contextual drift on one hand and contextual sparsity on the other is a challenging task. In BEL the range of relevant terms is calibrated empirically, by evaluating the performance of the contextual similarity score with different range sizes (i.e., for 10, 15, 20, ..., 155, after removing stop words and non-English terms). Figure 3.4 shows the average performance based on 10-fold cross validation on the above set of 1000 mentions. Both the precision and F1-measure achieve maximum when the relevant range contains 55 terms, which is also reported to be the optimal setting in [Gooi and Allan (2004)]. Thus, we define the size of the relevant range to be 55. Note that when a document contains less terms, BEL takes the whole text as the relevant range.



(a) Precision (b) Recall (c) F1-measure

Figure 3.4: Performance of the language model by increasing the size of the range of relevant terms. (in %)

By applying the bagging strategy, BEL can capture higher diversity in contextual information and provide a natural threshold of uncertainty of the ranking results. When none of the candidates is ranked as top candidate by the majority of the language model classifiers, the mention is considered to be non-YAGO entity. Here, to show the impact of our bagging strategy, we randomly pick 80% articles from our datasets and for different bagging sizes. Compare to a linking strategy that applies a single language model, the precision derived from majority voting is consistently higher. We run BEL 10 times for each bagging size on these articles (i.e., number of subsets derived from the range of relevant terms).

In Figure 3.5, the black horizontal line with precision 67.56% is the baseline precision of a single language model being applied to the range of relevant terms. The black dots denote the average precisions and the error bars show the corresponding standard deviation. As the figure shows, the precision increases with the increasing number of subsets but stabilizes after 65 subsets. At the same time, the standard deviation decreases. Considering the precision, efficiency and stability (in terms of the above standard deviation) of BEL, we propose 199 as the default setting. Note that the F1-measure of the majority voting slightly decreases, since the linking process is stricter and the recall

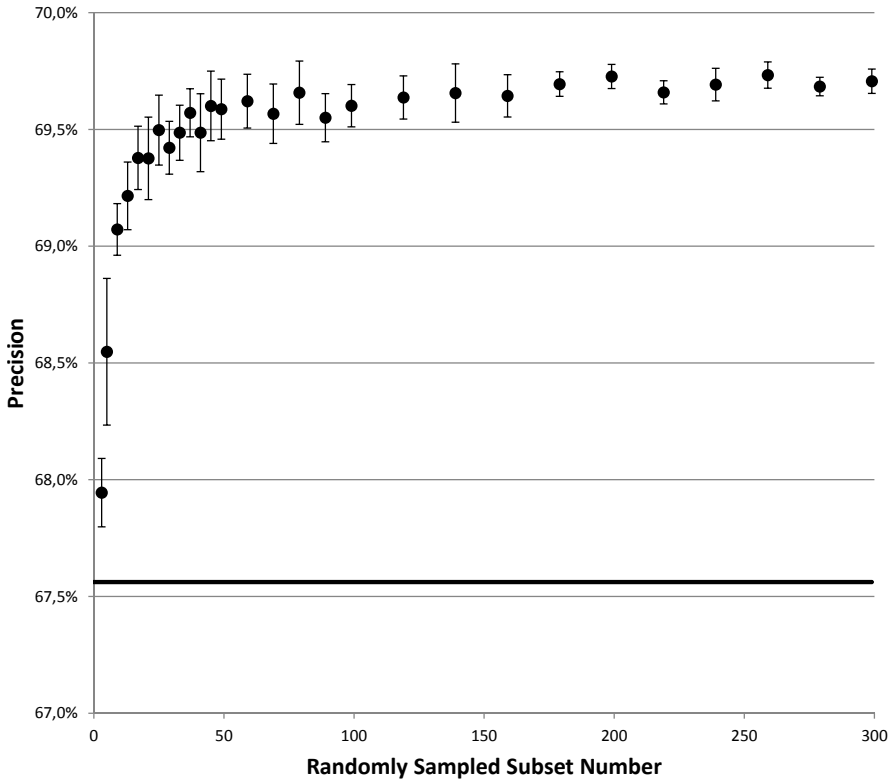


Figure 3.5: Performance of bagging strategy in precision for language model (in %).

decreases. However, in our opinion, it is better to suggest a mention to be not in the knowledge base than link it to a wrong entity. Furthermore, the experimental result shows that the impact of the bagging strategy on the increase of precision is consistently higher than its impact on the decrease of the F1-measure. For instance, by applying our default setting, which is 199, the precision increases from 67.56% to 69.73% while the F1-measure decreases from 75.82% to 75.32%)

3.5.4 Linear combination of scoring components

In our experiments, we linearly combine the "prominence" score and the contextual similarity score. As in our mathematical derivation (see Section 3.3), the combination parameter is set to 0.5, thus giving both scoring components the same weight. We have also observed that by fine-tuning the combination parameter to the specific type of corpus, the performance of BEL on the corpus can be improved. For example, we can give the "prominence" score a higher weight for news articles to get better performance, since most of the named entities occur in news are popular ones. However, the observed improvement has not always been significant. Therefore, in the following experiments,

we evaluate the general formalization of BEL's model where the linear combination parameter gives the same weight to both scoring components. This allows us to gain insights on the general performance of BEL on different datasets of different type of origin.

3.5.5 Evaluation

BEL is evaluated with respect to linking quality and efficiency. The employed evaluation measures and the results are presented in the following subsections.

Evaluation measures

For the quality evaluation, we have measured precision, recall, and the F1-measure of each of the above approaches on the mentioned datasets. A *true positive (tp)* is a mention that has been correctly linked to a YAGO entity. An incorrect linking is defined to be a *false positive (fp)*. Further more, a *true negative (tn)* refers to a mention that is correctly identified as an entity that does not occur in YAGO (i.e., non-YAGO entity). The remaining cases are defined as *false negatives (fn)*. Precision is then defined as $P = tp/(tp + fp)$, and recall as $R = tp/(tp + fn)$. The F1-measure is obtained from the harmonic mean of precision and recall as $F = 2PR/(P + R)$.

For the efficiency evaluation, we have measured the runtime (in seconds) of each approach on each dataset.

Linking quality results

The results of the quality evaluation are shown in Table 3.2, along with the corresponding confidence intervals, which are calculated by repeated random sampling of subsets containing 60% of the documents 30 times from each dataset. The variable $\overline{X}$ represents the mean of the respective score on the 30 subsets. For each dataset, the results computed on all documents are within these intervals with a confidence level of $1 - \alpha = 99\%$ according to the Student's t-distribution:

$$P(\overline{X} - \frac{t_{n-1,1-\alpha/2}s}{\sqrt{N}} \leq \mu \leq \overline{X} + \frac{t_{n-1,1-\alpha/2}s}{\sqrt{N}}) = 1 - \alpha \tag{3.6}$$

where the endpoints of the interval are $\overline{X} \pm \frac{t_{n-1,1-\alpha/2}s}{\sqrt{N}}$. This means that although some of the datasets are rather small, the confidence on the evaluation results to be located inside the interval is quite high.

As can be seen, BEL significantly outperforms all the other approaches on the CoNLL-YAGO dataset, especially on precision. Also, for the CUCERZAN dataset, the quality

Table 3.2: Evaluation results (precision, recall, and F1 in %).

	Method	Precision	Recall	F1
CoNLL-YAGO	LED	62.35 (-1.92,+0.25)	96.13 (-0.43,+0.27)	75.63 (-1.50,+0.18)
	AIDA-GRAPH	78.67 (-0.80,+1.23)	96.29 (-0.20,+0.64)	86.59 (-0.41,+0.82)
	AIDA-KORE	77.11 (-0.86,+0.80)	96.21 (-0.64,+0.25)	85.61 (-0.67,+0.47)
	BEL-PROM	68.37 (-0.89,+1.30)	97.40 (-0.25,+0.32)	80.30 (-0.61,+0.97)
	BEL	**81.40** (-1.33,+0.78)	95.72 (-0.38,+0.25)	**87.98** (-0.85,+0.46)
CUCERZAN	LED	63.47 (-0.40,+1.01)	96.94 (-0.11,+0.24)	76.72 (-0.28,+0.75)
	AIDA-GRAPH	81.30 (-0.57,+0.16)	94.64 (-0.28,+0.17)	87.47 (-0.40,+0.11)
	AIDA-KORE	81.35 (-0.83,+0.03)	97.31 (-0.25,+0.10)	**88.61** (-0.57,+0.03)
	BEL-PROM	73.92 (-0.53,+0.29)	98.83 (-0.11,+0.06)	84.58 (-0.37,+0.20)
	BEL	**82.37** (-0.31,+0.25)	93.46 (-0.71,+0.27)	87.56 (-0.35,+0.12)
KORE	LED	40.14 (-3.30,+0.88)	100.00 (-0.00,+0.00)	57.28 (-3.52,+0.79)
	AIDA-GRAPH	39.73 (-1.45,+2.29)	100.00 (-0.00,+0.00)	56.86 (-1.58,+2.24)
	AIDA-KORE	**66.67** (-2.29,+1.91)	94.95 (-0.87,+1.82)	**78.33** (-1.60,+1.48)
	BEL-PROM	31.29 (-1.83,+1.47)	100.00 (-0.00,+0.00)	47.67 (-2.23,+1.61)
	BEL	54.55 (-2.40,+2.53)	76.61 (-0.76,+2.20)	63.72 (-1.72,+2.08)
	BEL*	60.54 (-1.89,+1.25)	100.00 (-0.00,+0.00)	75.42 (-1.52,+0.93)

of BEL is comparable to that of AIDA-GRAPH and AIDA-KORE, and it significantly outperforms LED. Moreover, in terms of precision, BEL performs also on this latter dataset significantly better than the other approaches (i.e., from a statistical point of view). Together with BEL's impressive efficiency (see Section 3.5.5), the precision-related quality is a crucial scalability aspect, since when processing a high throughput of documents it is highly important that the produced linkings be rather correct.

For the KORE dataset, AIDA-KORE is the only approach that significantly outperforms BEL. However, BEL outperforms LED and AIDA-GRAPH. It should be noted that KORE is a very challenging dataset and that the AIDA-KORE approach has been specifically tailored to such datasets. Also note that although the AIDA-KORE algorithm shows a high linking quality in the experiments, it is the least efficient approach, since it performs complex joint reasoning over groups of candidate entities and mentions. In fact, in our experiments, we had to wait several hours, and sometimes even days (e.g., for the CUCERZAN dataset) for the evaluation results of this approach.

In comparison to a greedy linking strategy, where a mention is simply linked to the most prominent entity according to the "prominence" score, which is our baseline BEL-PROM, BEL performs much better on all three datasets. This fact highlights the importance of the contextual similarity component in the model.

By optimizing BEL's parameters for the KORE dataset, we were able to considerably increase precision, recall, and F1-measure, as shown in the last row of Table 3.2, while still maintaining a high efficiency. The parameters were set as follows: The parameter α was set to 0.625, thus increasing the weight on the language-model-based contextual score, the threshold on the number of generated candidates was increased to 125 (from 40 in BEL's original setting), the length of the original sliding window was decreased to 15 (from 20 in BEL's original setting), and the prediction of true negatives (i.e., non-YAGO entities) was turned off. In this version (i.e., BEL*), our approach considerably outperforms AIDA-GRAPH and LED. At the same time, the algorithm is still highly efficient.

Efficiency results

The lean algorithmic design of BEL (i.e., given by the linear combination of two light-weight models) enables a highly efficient linking process. For each dataset, Table 3.6 shows the runtime of each approach (in seconds). For the efficiency comparison, we used a Pentium 3.1Gh machine with 8GB of main memory. All datasets were loaded and processed by the algorithms in main memory. The indexes for the language models of YAGO2 entities, as well as the indexed YAGO2 knowledge base (that was used by all approaches for the linking) were maintained in a PostgreSQL 9.1 database.

For each dataset, Table 3.6 shows the runtime of each approach. Obviously, the joint reasoning strategy of AIDA-GRAPH comes at high efficiency costs; on all datasets it

Method	KORE	CoNLL-YAGO	CUCERZAN
BEL	30.02s	244.34s	657.44s
LED	17.70s	288.52s	552.26s
AIDA-GRAPH	615.35s	1,202.66s	2,897.43s
AIDA-KORE	>15h	>11h	>25h

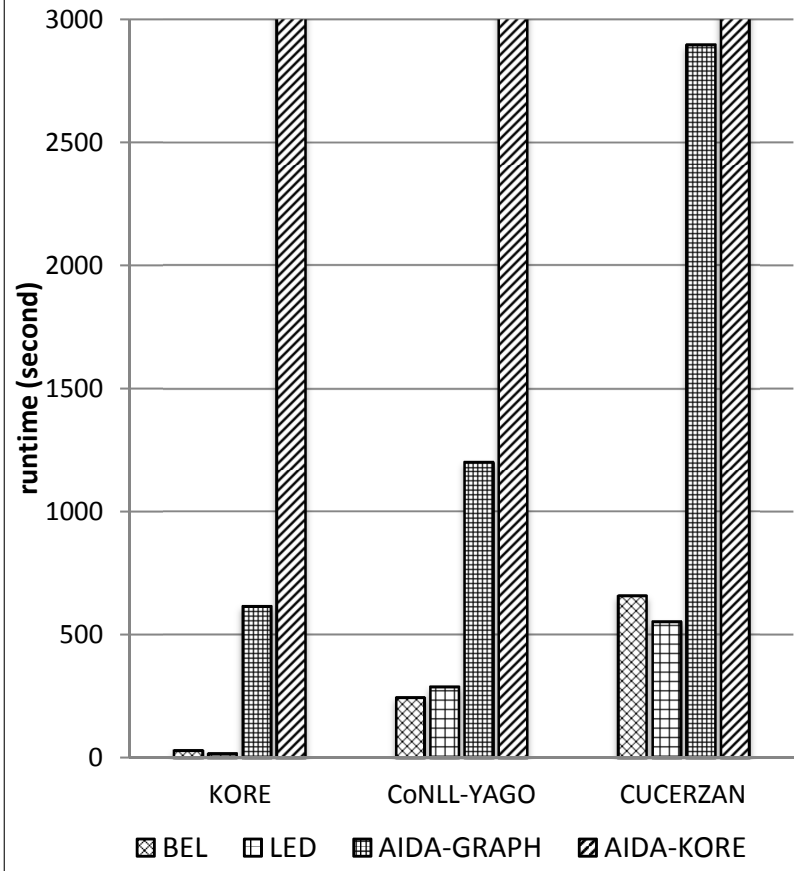


Figure 3.6: Efficiency comparison

has been outperformed by the other approaches. While LED is slightly more efficient than BEL on the KORE and CUCERZAN datasets, as the result presented in Table 3.2, it often pays a high cost in terms of quality, and BEL also outperforms it in terms of efficiency on the CoNLL-YAGO dataset. Note that the runtime of LED and BEL are both practically viable from a user's perspective. The AIDA-KORE approach, on the other hand, lacks practical viability, since it needs several hours to process even moderately sized datasets (e.g., approx. 11 hours for the CoNLL-YAGO dataset). For the CUCERZAN dataset, which is the largest one, although the F1-measure of AIDA-KORE is approximately 1% higher than BEL, one needs to wait more than 25 hours to get the result. Instead, BEL can finish the linking process in around 11 minutes.

3.5.6 Discussion

Both, the "prominence" score and the contextual score derived from the bagged language models have advantages and limitations; they are orthogonal in nature, and their individual strengths are manifested in different ways in the final decision of the algorithm. The value of the "prominence" score has a high impact on the final decision, when BEL is run on articles about famous people, organizations, locations, products, events etc. Typical examples of articles that contain such entities are news reports, scholarly articles containing encyclopedic knowledge, and product descriptions. In contrast, the bagged language models have a high impact on the final decision in cases where the occurring mentions are highly ambiguous but contain valid key information surrounding the mention. Examples for such articles can be found in all three datasets we have used.

Although in many cases, linking the mention to the most prominent candidate entity leads to the correct decision (e.g., "Einstein" refers most probably to the great physicist "Albert Einstein" and "Berlin" probably to the capital of Germany), this strategy is not reliable for many ambiguously used mentions. For example in one of the articles in the CoNLL-YAGO dataset, the "Australia National Cricket Team", which has a corresponding entity in YAGO, was often referred to as "Australia". For this example, among all competitors, only BEL and AIDA-KORE could find the correct linking; for many similar examples, BEL could establish a linking to the correct named entity by means of its bagged language models.

From the very beginning of BEL's design, a synergistic combination of the "prominence" score and the contextual language-model-based score, such that the advantages of both models are maximized, has been the key focus. Despite the very promising quality and efficiency results, we are confident that BEL's performance can be further improved by machine learning techniques to further boost the combined yield of both models.

3.6 Simplified named entity recognition and linking

As one of the fundamental steps of information extraction process, the result of many further applications depends heavily on recognizing and linking named entities reliably and efficiently in the first place. As introduced in Chapter 1, *Relation Extraction (RE)* is a typical follow-up task that based on the results of NER and NEL in a named entity mining system. According to the specific target relations, we can fine-tune the current NER or NEL approaches to adapt to the RE step. For instance, the RE approach that is introduced in the following chapter focuses on extracting business relations between companies. In this case, company mentions are the only type of named entities that need to be concerned. Therefore, we customized and simplified BEL to improve the overall performance, especially the efficiency. This simplified version allows an effective and efficient recognition and disambiguation of company mentions from large amounts of textual data.

First, as introduced in Section 1.1, the general NER task is to discover entity types, such as persons, locations, organizations. Therefore, most of the current existing NER approaches focus on recognizing the general entity types, which do not include the entity type *company* explicitly. The setting of the Stanford NER tagger [Finkel et al. (2005)], which is used for BEL, is also designed to recognize the basic types of named entities from text. Since *company* is a subset of *organization*, we recognize organizations by employing the Stanford NER tagger to generate a set of candidate company mentions. With this step, the ambiguity of mentions are significantly reduced, since all other types of named entities are already excluded in the candidate list.

In the next step we identify the companies from the set of recognized organizations via our simplified BEL approach. In this case, when a candidate mention cannot be linked to any company entity in a knowledge base, it is recognized as a NIL entity. To recognize and disambiguate the candidate company mentions, we choose DBpedia [Auer et al. (2007)], which is one of the most widely used knowledge bases to build a general dictionary of companies. DBpedia is a knowledge base that contains structured information which is derived from Wikipedia. In total there are 85,583 distinct company entities defined within DBpedia. We added the names of these entities as company names to our dictionary.

The same as the other types of named entities, companies are not always mentioned by its original name, but also by using diverse aliases. For instance, author can mention Microsoft as *MS*, ASUSTeK Computer Inc. as *ASUS*, East Japan Railway Company as *JR East*, and so on. Hence, if we directly match the original company mentions, we are missing all alias mentions. To improve the matching performance, we include the aliases of all companies contained in our dictionary. One straightforward way of finding these aliases is to automatically generate them from the original company names. However, as the previous examples show, aliases do not obeying any clear rules. Therefore, following the same strategy of computing the prominence score for BEL, we try to extract possible

company aliases based on web documents. In other words, a mention is an alias of a company only if it is used to refer to the same company in some articles.

Since DBpedia entities are derived from Wikipedia pages we can also extract aliases by using the information contained in Wikipedia, which is introduced in Section 2.3. A useful source is the large amount of hyperlinks contained in Wikipedia articles. Authors are required to link at least the first mention of a Wikipedia company entity to a corresponding article. An example of the mentions of companies in Wikipedia articles would be "*... in Germany's postwar auto market, sandwiched between [[Volkswagen|VW]] and [[Mercedes-Benz]]*". In the original sentence, *VW* is mentioned and hyperlinked to the company Volkswagen, while Mercedes-Benz is mentioned by its original name. We collect all mentions that link to one of the companies in the dictionary, or the name of a page that redirected to one of the Wikipedia company pages. In addition, we also record the frequencies of the aliases. After the extension, each company has on average 2.5 aliases (i.e., including its original name). Based on the frequency information we are now able to compute a prominence score, which denotes the probability that a mention m refers to a company c. This score is also an indication of the reliability of the final disambiguation result. Note that the dictionary can easily be extended by other sources, such as Freebase [Bollacker et al. (2008)] and YAGO [Suchanek et al. (2007)].Finally, we use the dictionary to recognize and linking companies by matching the textual mentions against it.

In the experiment, which we introduce in Section 4.6, mentions are greedily linked to the most prominent company in our dictionary. With this dictionary-based NER and NEL strategy we are able to recognize and disambiguate company mentions efficiently. Comparing to the normal setting of BEL, we leave out the component of the contextual similarity, which enables it to provide a large amount of tagged document for the relation extraction process. A simple evaluation of the recognition and linking result is presented in Section 4.6.3. Moreover, users can fine-tune the dictionary to their specific use case, e.g. by using only parts of the dictionary or extending it according to their requirements.

CHAPTER 4

Relation Extraction

In this chapter, we introduce an approach to extract business relations between companies from textual data. Our approach is inspired by Snowball [Agichtein and Gravano (2000)] that employed a bootstrapping strategy to extract organization and their corresponding headquarters. The basic idea is to provide a small set of entity pairs (i.e., a seed set) to generate candidate patterns, which are based on the context of the entity pairs. The most prominent patterns are selected according to a scoring function to extract new entity pairs that participate in the relation of interest. It uses the newly selected pairs to extend the seed set and iterates the process to get more patterns. Using Snowball as the starting point, we extended the system by introducing a key-phrase extraction strategy, which allows us to remove the irrelevant part of the context surrounding the company pairs. To determine the direction of asymmetric relations (e.g., ownership_of and supplier_of), we also propose a strategy that addresses the issue by leveraging the information contained in the seed set. We propose a generalized tuple and pattern evaluation strategy to select patterns and new seeds, which can be applied to extract the many-to-many relations that Snowball cannot handle.

The contents of this chapter are structured as follows: In Section 4.1, we introduce the motivation of extracting business relations. We introduce the overview of our approach in Section 4.2 and the detailed algorithm in Section 4.3. We present how we identify the direction of asymmetric relations in Section 4.4. We also introduce a holistic pattern identification strategy, which enables us to handle multiple relation types in Section 4.5. Finally, Section 4.6 shows our experimental results, which outperform the state-of-the-art approaches in terms of quality or efficiency. This chapter is based on the work [Zuo et al. (2017)]. The detailed introduction of related work can be found in Section 2.4

4.1 Business relations

Extracting structured data from text, and thus harnessing the valuable information on the web, and hidden in the vast amounts of other textual data, is a well-known and well-studied research area. As the text corpora and the kind of information to be extracted from them can vary greatly, many research works have focused on specific types of information, on specific corpora, on specific application domains, on specific languages, or any combination of the above. In this paper, we regard the problem of extracting *relations* of several specific types among *companies* from *news articles*.

Many tasks, such as building business networks, predicting risks, or valuating companies, can significantly benefit from accurately extracting relations between companies. Imagine a scenario in which Dell wants to acquire EMC. Dell plans to finance the deal by taking out a loan. The chosen bank has to decide whether to award the loan based on the careful assessment of the risk associated with this transaction. The traditional insurance principle for risk mitigation relies on the assumption of independence of individual contract values, which has been proven to be quite wrong, e.g., during the financial crises of 2008/2009. However, it is a difficult task to quantify the dependencies, due to the lack of relevant historic data. With the explosive growth of the textual data on the web, it becomes possible to discover such dependencies by extracting business relations and building up a company network. Back to the previous example, by analyzing the network structure of both companies, the bank might reach the conclusion that the risk of granting a loan is too high, because of many of EMC's subsidiaries, as given by the relation network, are struggling. With this knowledge, the bank might award a smaller or no loan at all or propose a higher interest rate.

To build up a business network between companies, it is critical to extract business relations reliably. Companies often connect to each other via the activities in which they participate. Business relations represent a subset of these activities; examples include ownership_of, partnership_with, supplier_of, and so on. Only very few of them can be found in structured knowledge base like Freebase [Bollacker et al. (2008)] or semi-structured data like Wikipedia infoboxes – a substantial amount of relations is hidden in unstructured data sources. Aggravating this situation, both Freebase and infoboxes contain only the major subsidiaries of some companies (i.e., ownership_of relation). Other relations, such as partnership_with or supplier_of, are not covered.

As an example, Figure 4.1 shows an excerpt of a business network that was created based on selected ownership_of relations. The orange dashed lines denote relations that can be extracted from news articles, while the gray dotted ones depict the connections derived from Wikipedia infoboxes. In this excerpt, only three links are identified from both sources – depicted by green lines. Although much work on relation extraction has been done, it has mainly focused on extracting the most frequently mentioned relations, such as the nationality of a person, the birthplace of a person, or the headquarters of an organization. The subject of business relations between companies has not been

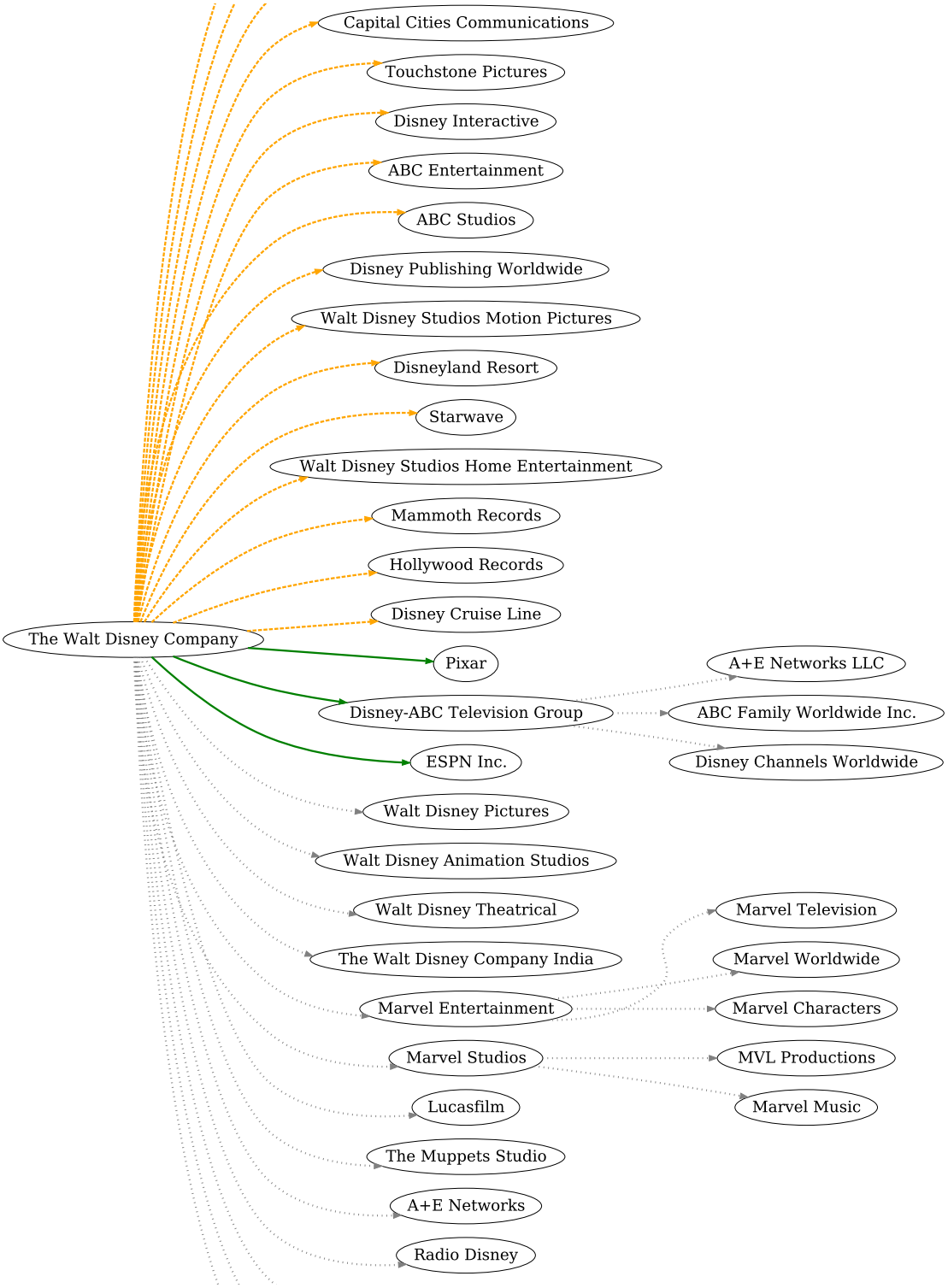


Figure 4.1: A partial tree of the subsidiaries of Walt Disney. The relations that can be extracted from various sources are denoted by different line styles as follows: Orange dashed line: news articles; Gray dotted line: Wikipedia infoboxes; Green line: both sources.

addressed.

Given a corpus ofirst unstructured textual data, we aim to (1) discover whether two co-occurring companies *participate* in a business relation, (2) identify the *type* of the relation, and (3) in the case of an asymmetric business relation, determine its *direction*.

The task of business relation extraction is challenging due to the complex nature of the relations between companies. First, multiple types of relations can exist between two companies simultaneously. Samsung as one of the biggest competitors of Apple is also the supplier of displays for Apple's products. Moreover, as an example of resolving the direction of asymmetric relations, such as the ownership_of relation, consider that Walt Disney owns ABC Studios but not the other way around. Being able to derive the direction of the relations successfully is of vital importance for many following tasks.

The Snowball system addresses the general problem of relation extraction [Agichtein and Gravano (2000)], and our work is based in parts on its general idea. It takes a small set of entity pairs as a seed set and generates candidate patterns that are based on the context of these pairs. Subsequently, the most prominent patterns are selected according to a scoring function and used to extract new entity pairs that participate in the target relation. In the end, the newly selected pairs are added to the seed set and the process repeats to generate more patterns. However, Snowball functions only correctly if there is a one-to-many relation between the participating entities, e.g., in the headquarter_of(Microsoft, Redmond) relation, Microsoft has exactly one headquarter. Business relations do not adhere to this characteristic, which is the reason Snowball is unable to solve the problem at hand.

We extend the Snowball idea by introducing a key-phrase extraction strategy, which allows us to remove irrelevant parts of the context surrounding the company pairs. For example, in a snippet "*... Booklamp, an important unit of Apple ...*", the key-phrase to describe the relations between Booklamp and Apple is "unit of", the rest terms in the contexts are unimportant for extracting this type of relations. Furthermore, to determine the direction of asymmetric relations, we propose a process that leverages information contained in the seed set. As mentioned in the original paper of Snowball [Agichtein and Gravano (2000)], due to the limitation of confidence score designed in the Snowball system, their approach cannot deal with many-to-many business relations. Therefore, we propose a generalization of their tuple- and pattern-evaluation strategy by specifying a new selection method to select patterns and new seeds. We further define a holistic pattern identification strategy, which enables us to extract multiple relation types simultaneously.

In summary, we propose a system to perform *(directed) relation extraction (RE) between companies* from textual data. Addressing this problem, we present a novel, semi-supervised relation extraction method, which requires only a minimum amount of manually specified company pairs to efficiently extract new ones that belong to the same target relation. Additionally, we provide a straightforward solution to identify the di-

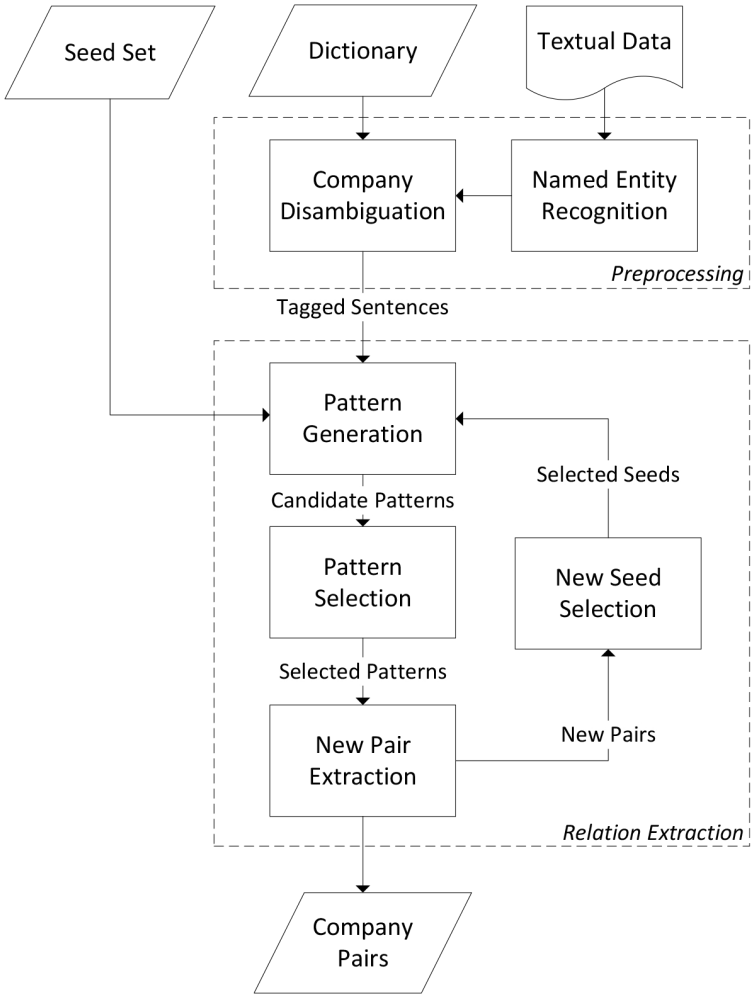


Figure 4.2: Pipeline of business relation extraction approach

rection of asymmetric relations reliably. We show that our approach is superior to more advanced distant learning approaches for the particularly difficult case of many-to-many relations.

4.2 Overview of approach

Figure 4.2 gives a high-level overview of our relation extraction approach, which is inspired by the basic pipeline of Snowball [Agichtein and Gravano (2000)]. The task is to extract company pairs from unstructured data that participate in a predefined relation.

Given some textual data, such as news articles, encyclopedic articles, or company websites, we use a seed set which consists of multiple company pairs that occur as

members of a particular relation to initialize our bootstrapping approach. We define a seed set as a set of manually collected company pairs, which characterize a predefined relation type of interest. The direction of an asymmetric relation is specified by always keeping the seed pairs in a forward direction.

Given the input data, we first perform an initial company recognition step using Stanford NER [Finkel et al. (2005)] approach. We then use the simplified version of BEL, which introduced in Section 3.6, to link the mentions of companies to their corresponding entity in our dictionary. In the next step we tag the disambiguated companies in our input texts, which means that if a mention is recognized to refer to a company in our dictionary, we label this mention with the corresponding company name (see Section 3.6). In our experiments, we follow the intuition that if a company pair from the seed set co-occurs in the same sentence, it is likely that the context characterizes the relation specified by the seed (see Section 4.3.2). Therefore, the sentences that contain two or more distinct companies are selected as the input for the relation extraction phase. An example tagged sentence is "*... with most of that coming from [[Verizon Communications|Verizon]]'s acquisition of [[MCI Inc.|MCI]] last year*". In this example, the original mention "Verizon" is disambiguated as *Verizon Communications*, while *MCI Inc.* is disambiguated as "MCI".

From the contexts surrounding company pairs we generate possible candidate extraction patterns that are likely to represent the relation of interest. These patterns are generated by using a key-phrase extraction strategy (see Section 4.3.2). For example, suppose we are interested in the ownership_of relation and the company pair (Verizon Communication, MCI Inc.) is contained in the seed set, then a candidate pattern $pattern = \langle$COMP1, COMP2, acquisition of, $\rightarrow\rangle$ can be generated. With our key-phrase extraction strategy, the key-phrase "acquisition of" is extracted from the original context. The last element in *pattern* describes the direction of the ownership_of relation. Here the direction is marked as a forward direction (see Section 4.4). After generating the list of candidate patterns from the seeds, we select the most promising ones according to the measurements introduced in Section 4.3.3.

During the extraction process, we use the selected patterns to discover new company pairs from the input. For example, if the *pattern* is selected, we can extract a new company pair (The Walt Disney Company, Pixar) from a sentence like "*... after [[The Walt Disney Company|Disney]]'s $ 7.4 billion acquisition of [[Pixar|Pixar Animation Studios]] ...*". Afterward, we select the most prominent new extracted pairs according to our scoring function to extend the seed set (see Section 4.3.4). We then iterate the procedure of extracting new patterns using the extended seed set.

The iteration process terminates when no more company pairs can be selected as seeds, or the iteration number reaches a predefined limit. The company pairs that are extracted based on the current set of patterns are considered to have the same semantic meaning as the relation of interest. This is defined by the company pairs in the initial seed set. Our evaluation shows that this is indeed almost always the case, regardless of

the choice of seeds.

4.3 Business relation extraction

This section introduces our semi-supervised relation extraction strategy. The general idea is that, given an initial seed set that represents some type of relation, our approach iteratively extracts new company pairs that participate in a relation of interest. In Section 4.5 we address the case of extracting multiple relation types simultaneously.

4.3.1 Overview of algorithm

An overview of the extraction strategy is shown in Algorithm 2. As input we provide the sentences (**S**) that are annotated based on the company extraction results and a seed set ($\mathbf{Pair_{seed}}$), which consists of several manually collected company pairs. All company pairs in the seed set participate in a relation r. To handle asymmetric relations we define that each pair also holds a positive direction.

First, an initially empty list for storing extraction patterns $\mathbf{Pattern_{selected}}$ is created. Lines 1 to 8 generate candidate patterns from the sentences. To generate these patterns we determine for each sentence if it contains a company pair from the seed set and condense the context between the two co-occurring companies down to a single key-phrase. Using the key-phrase we generate the pattern and add it to the candidate pattern list $L_{pattern}$ (lines 4, 5).

According to one of the strategies introduced in Section 4.3.3, the most prominent patterns in $L_{pattern}$ are selected to extend $\mathbf{Pattern_{selected}}$ (line 9). The patterns in this set are now used to find new company pairs that participate in r. The newly extracted pairs form the set $\mathbf{Pair_{new}}$ (line 10). According to the selection strategy presented in Section 4.3.4, we now select new seed pairs to extend the current seed set (line 11). If there are additional pairs that can be added to current seed set and the number of iterations does not reach the predefined limit, we iterate the procedure (**go to** line 1) and apply the extended seed set to extract new patterns (lines 12-14). When the process terminates, our system returns all newly extracted company pairs $\mathbf{Pair_{new}}$ as our extraction result (line 16).

4.3.2 Pattern generation

Generating the extraction patterns represents a crucial step in our approach. The context surrounding a company pair represents the primary source to identify relations occurring in textual data. To capture the key information that describes the relation between two

Algorithm 2 Basic Algorithm of relation Extraction

Input: Tagged sentences $\mathbf{S} = \{s_1, s_2, \ldots\}$, each sentence $s := (cont_l, c_i, cont_m, c_j, cont_r)$; $\mathbf{Pair_{seed}} = \{pair_1, pair_2, \ldots, pair_n\}$, $pair := (c_i, c_j), i \neq j$.
Output: new company pairs $\mathbf{Pair_{new}} = (pair_1, pair_2, \ldots)$

1: **for** each sentence $s_i \in \mathbf{S}$ **do**
2: **for** each pair $pair_i \in \mathbf{Pair_{seed}}$ **do**
3: **if** $match(pair_i, s_i)$ **then**
4: $pattern_{cand} = generatePattern(pair_i, s_i)$
5: $L_{pattern}.add(pattern_{cand})$
6: **end if**
7: **end for**
8: **end for**
9: $\mathbf{Pattern_{selected}}.add(selectPatterns(L_{pattern}))$
10: $\mathbf{Pair_{new}} := extractNewPairs(\mathbf{S}, \mathbf{Pattern_{selected}})$
11: $\mathbf{Pair_{newseed}} := selectNewSeeds(\mathbf{Pair_{new}})$
12: **if** $\mathbf{Pair_{newseed}} \nsubseteq \mathbf{Pair_{seed}}$ or $\#Iteration < \tau$ **then**
13: $\mathbf{Pair_{seed}} = \mathbf{Pair_{newseed}} \bigcup \mathbf{Pair_{Seed}}$
14: **go to** 1
15: **else**
16: return $\mathbf{Pair_{new}}$
17: **end if**

companies we extract the most determining phrases from the context as a key-phrase. This key-phrase is then used to generate a pattern.

Candidate pattern

An extracted *pattern* includes two company variables COMP1 and COMP2, the key-phrase extracted from the context in between these companies, and a direction. We explain each of these parts in the following. For example, from the sentence "... `YouTube, the video-sharing Web site owned by Google, ...`" we can generate the pattern ⟨`COMP1`, `COMP2`, owned by, ←⟩. By applying this pattern to the example sentence we get the following instantiation of the pattern ⟨`YouTube`, `Google`, owned by, ←⟩. Considering the direction field of the extracted instance it is now easy to see that Google is the owner of YouTube.

Key-phrase extraction

The quality of a pattern depends on the key-phrase it contains. A good pattern should satisfy two criteria. First, it should specifically characterize a single type of relation, which in turn improves the precision of the extraction result. Second, patterns should

be as general as possible to extract more new company pairs. For this reason, it is beneficial to generalize the context and keep only a key-phrase. The key-phrase should be as compact as possible while maintaining the semantic meaning of the context. We argue that our case of detecting relations is complicated since considering the language diversity and accuracy in news, journalists are used to introduce the same company relations using different writing styles spanning a relatively large context. This can be shown using the excerpt "... News Corporation, which owns a minority interest in DirecTV". In this sentence, we can easily figure out that News Corporation is the owner of DirctTV by finding the verb "owns" in the intermediary context. If we now use the entire context (i.e., ", which owns a minority interest in") between the two companies as a pattern to extract additional company pairs, we would only find very few of them since the pattern is not general enough. The problem can be solved by extract the key-phrase "owns" that defines the ownership relation between the two companies is the verb . As a result it is enough to identify the ownership relation by simplifying the original sentence into "New Corporation owns Direct TV".

Therefore, we develop a key-phrase extraction strategy to automatically extract the most determining phrases from the intermediary context. Intuitively, the key information in a sentence is often conveyed by verbs or nouns. An experimental result also proves this intuition, which is introduced by Banko et al. (2008). The result shows most of the binary relations that are mentioned in English texts are indicated by four types of phrases, which cover over 86% of the cases. These key-phrase types are "Verb", "Noun+Prep.", "Verb+Prep.", and "Infinitive", which are all located in the context between two named entities.

To extract key-phrases from contexts, we apply the Stanford log-linear Part-Of-Speech (POS) Tagger[1] to label all tokens in the sentence with POS tags. Based on these tags, we keep the phrases that match any of the four types above. Note that, "Verb+Prep." and "Infinitive" have higher priority than "Verb", which means "Verb" is used to generate patterns only if we cannot match it with one of the other types. For example, a piece of context "... *acquisition of...*" is given, the extracted key-phrase is "acquisition of", but not the verb "acquisition" alone. If multiple key-phrases occur in a context, we abandon it, because the pattern generated by this case is usually not general enough. Note that we exclude the verb "to be", because it usually does not indicate any business relation between companies.

4.3.3 Pattern selection

As the result of pattern generation, a set of candidate patterns is generated based on the seed set in each iteration. However, not every candidate pattern is suitable to be used during the extraction process. Patterns that do not represent the relation of interest should not be selected for following iterations. Considering the inner workings

[1] http://nlp.stanford.edu/software/tagger.shtml

of our extraction strategy, the quality of the final results depends on the patterns that are selected to extract new company pairs. Therefore, it is important to keep only the patterns represent the relation of interest, while filtering out unfavorable patterns. Otherwise, once an unfavorable pattern is selected, the effects of this selection propagates from iteration to iteration. We introduce two strategies to select patterns, one is based on a Hit score, which represents the frequency of an extraction pattern that can be generated by the seed pairs. The other is based on a Coverage (Cov) score, which shows the percentage of seed pairs that can generate the corresponding extraction pattern.

Hit score

Building on the intuition that patterns that frequently match company pairs in the seed set are likely to be representative ones, we introduce a Hit score for each pattern as follows,

$$Hit(pattern|\mathbf{Pair_{seed}}, \mathbf{S}) = \sum_{pair_i \in \mathbf{Pair_{seed}}} \sum_{s_j \in \mathbf{S}} [match(pair_i, p, s_j)] \tag{4.1}$$

Thus Hit is defined as the summation of how frequently a *pattern* matches a company $pair_i \in \mathbf{Pair_{seed}}$ in the set of input sentences $\mathbf{S}$.

A pattern with a high Hit score frequently co-occurs with company pairs in the seed set, which means that the corresponding key-phrase is more likely to represent the relation of our interest. Based on this observation we use the Hit score to develop a pattern selection method: Given a list of candidate patterns that are sorted in descending order by their respective Hit score, we greedily select the top ranked patterns to extend the set of the current extraction patterns. Note that, among the top-k patterns, only those that are not already contained in the extraction pattern set of the previous iterations are added.

Coverage score

Instead of selecting patterns according to the absolute matching frequency of company pairs in the seed set, we introduce another selection strategy to select extraction patterns from the candidates. The idea is that a pattern that represents the relation of interest should frequently be used in the context between two companies to describe just this relation. This implies that the probability that such a pattern co-occurs with the company pairs in the seed set is high. Here the ideal case is that the pattern occurs together with all seed pairs at least once. If the pattern can be extracted by using only one of the seed pairs, it is either too specific or it describes some other type of relation between the corresponding companies.

To deal with this issue, we introduce a Cov score to evaluate the candidate patterns. It represents the percentage of company pairs from seed set that are able to generate this pattern. We call this ratio Cov score of the corresponding pattern, which is calculated as follows,

$$Cov(pattern|\mathbf{Pair_{seed}}, \mathbf{S}) = \frac{\sum_{pair_i \in \mathbf{Pair_{seed}}} [\sum_{s_j \in \mathbf{S}} [match(pair_i, p, s_j)] > 0]}{|\mathbf{Pair_{seed}}|} \tag{4.2}$$

The Cov score of a pattern equals 1 when all seed pairs match the corresponding pattern. Together with the Cov score, we define the threshold τ, which is used to filter out a subset of patterns during the selection process. All patterns that have a Cov score greater than τ are selected for extending the set of selected patterns.

4.3.4 New seed selection

We introduce a similar strategy for selecting newly extracted company pairs to extend the seed set. Using the selected patterns, we compute a Hit score for each of the extracted company pairs. Given these Hit scores, we are able to select the top-k pairs, which we rank according to their Hit scores. We also compute the Cov score of an extracted company pair, which is the percentage of selected pattens that match the company pair in the text. In a similar fashion to the pattern selection described in Section 4.3.3, we select a company pair that has a greater Cov score than a given threshold τ and use it to extend the seed set.

4.4 Direction of relations

In Section 4.1 we introduced the challenge of determining the direction of asymmetric business relations. Compared to the extraction of symmetric relations, extracting asymmetric relations, such as supplier_of and ownership_of requires not only the extraction of the company pairs participating in the relation but also the detection of its correct semantic direction. For example, regarding the ownership_of relation, it is of crucial importance to know which company is the owner of another one; supply chains can be derived correctly, only if their direction is known.

Previous work, such as Snowball [Agichtein and Gravano (2000)], naturally avoids this direction problem, since they focus on relations that relate two objects of different entity types (i.e., organization and location). However, in our case, the entities on both sides of the relation are of the same type (i.e., company). In [Zhu et al. (2009)], a similar challenge presents itself. Here an entity of type person e_1 is the husband of e_2, which

automatically conveys information about e_2 being the wife of e_1. They have to manually adding new rules, such as $IsHusband(e_1, e_2) \Rightarrow IsWife(e_2, e_1)$, during their iterations.

To efficiently handle asymmetric relations, we introduce an elegant strategy to automatically classify the direction of newly extracted relations. The idea is to include the direction information already in the seed set. If we want to extract an asymmetric relation, the company pairs in the initial seed set must be specified by also providing the direction of the relation. E.g., in case of the ownership_of relation, we specify a forward direction, denoting that the first company is the owner of the second.

Given this directed seed set, we can identify the direction of the generated patterns as follows: Every time a pattern co-occurs with one of the seed pairs, we check whether two companies are mentioned in the same order as the seed pair. If this is true, the pattern is annotated with a forward direction. Otherwise, we annotate it with a backward direction. Finally, after the pattern generation process is finished, the final direction of a pattern is derived by assigning the direction of the pattern that was more frequently marked. A peek at Table 4.2 in the evaluation section shows some examples of (mostly correctly) determined directions of patterns.

By using the directional information of a selected pattern, we can also identify the direction of the relation existing between two companies. For example, given the sentence "*Youtube is a unit of Google.*", suppose we have a pattern containing the key-phrase "unit of", we deduce that it belongs to the ownership_of relation, and correctly assign it a backward direction. Using this information we then extract the company pair in the order Google and Youtube, which means Google is the owner of Youtube.

4.5 Holistic pattern identification

With our semi-supervised business relation extraction approach, we can independently extract different relation types by providing multiple initial seed sets each characterizing one type of relation. In this section, we introduce a holistic pattern identification strategy to deal with multiple relations simultaneously by assigning each pattern to a single relation.

As mentioned in Section 4.1, different types of business relations can exist between two companies at the same time. The patterns generated from the seed set do not always represent the desired relation type. Even worse, once a pattern that represents an undesired relation type is selected, the following iterations can be negatively influenced in a way that they yield more and more irrelevant patterns, which leads to incorrect extraction results comparable to a semantic drift in pseudo-relevance feedback methods. We introduce a holistic extraction strategy to reduce the semantic drift problem. The idea is to extract multiple types of relations simultaneously and assign each pattern that is generated for multiple relation types exclusively to only one single type.

We follow the intuition that each pattern characterizes one kind of relation. As a simple example, consider the relations ownership_of and partnership_with. A pattern with the key-phrase "owned by" should only be used to extract the ownership_of relation, since it is semantically closer to this relation than it is to the partnership_with relation.

We implement the core idea of our holistic pattern identification strategy by using the *Cov* score introduced in Section 4.3.3. In case the same pattern is generated for multiple relation types, we compare the patterns *Cov* score for each of the possible relation types and exclusively assign it to the relation type that yields the highest *Cov* score. A preliminary experiment to show the effect of this holistic strategy is presented in Section 4.6.3

4.6 Experiments

In this section we present the experimental results of our relation extraction strategy. To show the performance of our approach, we focus on the extraction of an asymmetric relation (i.e., ownership_of) from the New York Times news corpus. As a fundamental business relation type, ownership_of relations are one of the most frequently mentioned ones in news articles.

4.6.1 NYTimes corpus and seeds

The full New York Times corpus contains 1,855,658 news articles, spanning a period of 20 years from Jan. 1987 to Jun. 2007. Every article can have multiple section labels attached to it, such as "Arts", "Education", and "Sport". We observed that about 74% of all company pairs within a sentence occurred in the "Technology" and "Business" sections. Thus, we reduced our corpus to articles with at least one of those two labels, and our final corpus, called NYTimes from now on, consists of 359,459 articles. As we have introduced in Section 4.3, our approach extracts relations based on sentence-level information, which means a sentence can only contain a relation if there are at least two distinct company mentions in it, based on named entity linking results. Our NYTimes corpus yields 213,217 sentences that fulfill this condition.

An initial seed set serves as the input for our approach and predefines the relation we would like to extract. We investigated two different seed sets to evaluate their influence on the results. To this end we generated a list of distinct company pairs that co-occur in the NYTimes corpus and sorted it in descending order by co-occurrence frequency. We randomly selected company pairs from the top-100 ranked one until we find 5 pairs that participate in the ownership_of. The 5 pairs build up the *FreqSeed* seed set. Following this random selection strategy, we also generated a seed set called *InfreqSeed* from the top-1000 company pairs. The company pairs contained in those two seed sets are shown in Table 4.1. Keep in mind that seed selection and our evaluation is based on a corpus

dating from 1987 to 2007, so not all relations still hold today.

Table 4.1: Seeds randomly selected from Freq- and InFreq-occurring company pairs

FreqSeed		
Parent	**Subsidiary**	**#Co-occurrence**
General Electric	NBC Sports	895
Viacom	Viacom Media Networks	526
Ford Motor Company	Jaguar Cars	371
AOL	Netscape	364
Time Warner	Turner Broadcasting System	348
InFreqSeed		
Parent	**Subsidiary**	**#Co-occurrence**
General Motors	Saturn Corporation	192
Chrysler	American Motors	163
The Walt Disney Company	ESPN Inc.	145
Interpublic Group of Companies	Campbell Mithun	105
Investcorp	Saks Fifth Avenue	100

4.6.2 Results of pattern generation

We first show automatically generated patterns. Based on the two randomly generated seed sets we applied our approach to extract new company pairs that are also members of the ownership_of relation. Table 4.2 shows the key-phrases of the selected patterns that are generated by using *FreqSeed* and *InfreqSeed*. In this experiment, we applied the *Hit* score in each iteration for selecting the top-10 candidate patterns and the respective company pairs, see (Sections 4.3.3 and 4.3.4). The first column shows the key-phrases of the selected patterns, which were extracted from the context. By using either *FreqSeed* or *InfreqSeed*, the extraction process terminates after three iterations resulting in 14 selected patterns. These patterns are sorted in descending order by their *Hit* score. We also include the ranks of the patterns per iteration to show the changes that occur from iteration to iteration.

Further, Table 4.2 shows that most of the automatically generated key-phrases are typical phrases, frequently used to describe an ownership_of relation. Already in the first iteration our approach can generate representative patterns. Differences between the two sets of generated patterns can be mainly observed in the tail. It is worth noting that these differences already occurred during the first iteration, but did not increase in the following iterations. After three iterations, the two pattern sets are quite similar to each other in terms of both content and ranking. Thus, our approach is not particularly sensitive towards the chosen seed set (we made similar observations for various other seed sets).

The last column in Table 4.2 contains the extracted direction of the patterns, as determined by the strategy introduced in Section 4.4. Only two of the 16 directions

Table 4.2: Key-phrases of selected patterns for extracting ownership_of relation

Extracted Patterns (key-phrase)	Rank (Iteration)						Direction
	FreqSeed			InfreqSeed			
	1	2	3	1	2	3	
unit of	1	1	1	4	1	1	←
parent of	–	4	2	–	4	2	→
owned by	2	2	3	1	3	4	←
part of	4	3	4	2	2	3	←
division of	5	5	5	3	5	5	←
owns	3	6	6	7	6	6	→
(parent) company of	–	–	7	–	9	8	→
acquisition of	7	7	8	6	7	7	←
subsidiary of	–	8	9	–	8	9	←
owner of	–	–	10	–	–	10	→
including	9	9	11	–	–	–	→
include	8	10	12	–	–	–	→
bought	10	11	13	9	11	12	→
acquired	6	12	14	5	12	13	→
buy	–	–	–	10	10	11	←
bought by	–	–	–	8	13	14	←

are incorrect, both for an interesting reason: The addition of the company pair (Time Warner, AOL), which was selected as a new seed pair after the first iteration, caused the two marked incorrect directions (i.e., "acquisition of" and "buy"). In the year 2000, AOL's acquisition of Time Warner induced a merge structure. However, in the following years, the situation changed, and AOL became a unit of Time Warner. In this special example, the acquisition did not make AOL the owner of Time Warner, which caused the two incorrect directions in Table 4.2. Although the direction of these two patterns is incorrectly classified, most directions of the newly extracted company pairs, including Time Warner and AOL, are identified correctly as the statistics in Section 4.6.3 show. This is possible since the direction of newly extracted company pairs is determined by multiple patterns.

4.6.3 Quality of extraction results

To evaluate the quality of the actual business relations that we extracted, we applied our approach using different settings for both pattern and seed selection to verify the extraction result. We conducted experiments with the *Hit* and *Cov* scores strategies introduced in Section 4.3. To show the effect of our key-phrase extraction strategy, we also executed our algorithm without using this strategy. In other words, we employed the original context to generated patterns, which is similar to previous work, e.g., Agichtein and Gravano (2000); Brin (1999). As a baseline, we greedily select the most frequently co-occurred company pairs to check how many of them are in an ownership_of relation.

For evaluation, we had to manually checked relations between company pairs, because no gold standard with known business relations is available. The manual annotation of relations can be quite tedious and time-consuming as many relations need to be confirmed by searching for evidence on the web. Considering a large number of distinct company pairs contained in the NYTimes corpus, we designed two experiments to evaluate the extraction results, while at the same time reducing the labeling workload.

Table 4.3: Manually verified top-100 most frequently extracted company pairs.

Strategy	**P@100**	**Error Type**			
		Rel.	**Dir.**	**Com.**	**Sem.**
Baseline	36.0%	55.0%	0.0%	3.0%	6.0%
Hit@10 w/o KP	18.0%	74.0%	2.0%	2.0%	4.0%
Hit@5	90.0%	5.0%	0.0%	4.0%	1.0%
Hit@10	89.0%	4.0%	2.0%	4.0%	1.0%
Hit@15	88.0%	4.0%	2.0%	4.0%	2.0%
Cov w/o KP(τ0.6)	21.0%	70.0%	2.0%	3.0%	4.0%
$Cov(\tau = 0.5)$	87.0%	4.0%	2.0%	4.0%	3.0%
$Cov(\tau = 0.6)$	87.0%	4.0%	2.0%	4.0%	3.0%
$Cov(\tau = 0.7)$	90.0%	4.0%	0.0%	4.0%	2.0%
$Cov(\tau = 0.8)$	88.0%	4.0%	2.0%	4.0%	2.0%
$Cov(\tau = 0.9)$	90.0%	5.0%	0.0%	4.0%	1.0%

The design of our approach is mainly concerned with achieving a high precision value, because we aim to use it in the context of risk-analysis, which has only a small tolerance for incorrectly extracted information. Therefore, the first experiment is set up to evaluate the precision of our extraction results. We manually examined the top-50, top-100, and top-200 most frequently extracted company pairs from each result set produced by our algorithm with different parameterizations.

Table 4.3 presents the evaluation results using the *FreqSeed* seed set to extract the ownership_of relations. As this table shows, by applying *Cov* ($\tau = 0.7$) score, 90% of the top-200 extracted company pairs indeed participate in the ownership_of relation. The performance of our approach, excluding the key-phrase extraction strategy is also presented in Table 4.3. The result shows the significant effect of including key-phrase extraction strategy. In comparison to the baseline, our approach can produce much better results.

Apart from the precision measure, we also present a detailed error analysis based on the top-200 extracted company pairs: The first error type is that company pairs that do not participate in an ownership_of relation are extracted (Rel.). Another error case is that our approach extracted the correct company pair, but failed to identify the correct direction (Dir.). A third error case is caused by recognition or disambiguation errors made by the preprocessing steps (Pre.). An incorrect result can also be due to misinterpretation of the semantics (Sem.). E.g., one company finally canceled the plan of acquiring another one, such as the abandoned merger between EMI and Time Warner.

Such events are covered by a series of New York Times articles, but our approach was unable to capture the final cancellation of the deal successfully. As a result shows, only around half of the incorrectly extracted relations (i.e., Rel. and Dir.) are caused by our RE strategy. By employing the *Cov* score, it is more likely the select patterns that can be generated by multiple seed pairs, so that most of the results of direction classification are correct.

To evaluate the recall performance of our approach, we manually verified the top-200 most frequently co-occurring company pairs. 61 out of 200 company pairs were identified as participating in an ownership_of relation. These manually labeled company pairs are also used as a baseline to show the results that are generated by naively extracting the ownership_of relations according to the co-occurrence frequency of company pairs. Please note that, although those labeled pairs are mentioned together in the NYTimes corpus with a relatively high frequency, the corresponding contexts could contain no information about the aimed relation.

The recall performance based on the top-200 co-occurring company pairs is shown in Table 4.4. Here, 5 out of 61 labeled company pairs are included in the seed set, therefore the computation of our recall is based on the remaining 56 company pairs. The results show that the top-200 frequently extracted company pairs already cover most of the labeled pairs. By using the *Cov* score with a threshold of $\tau = 0.5$, our approach achieves the best recall, but also includes more incorrect company pairs in top-200 ranked ones. Increasing the threshold to $\tau = 0.7$ produces a balanced result in terms of precision and recall.

Table 4.4: Evaluation results by top-200 most frequently co-occurring company pairs (56 correct pairs + 5 seed pairs).

Strategy	**Correct**	**Incorrect**	**P**	**R**
Baseline	56	139	28.72%	100.00%
Hit@10 w/o KP	36	111	24.49%	64.29%
Hit@5	30	8	78.95%	53.57%
Hit@10	38	14	73.08%	67.86%
Hit@15	41	17	70.69%	73.21%
Cov w/o KP(τ0.6)	38	108	26.03%	67.86%
$Cov(\tau = 0.5)$	49	27	64.47%	87.50%
$Cov(\tau = 0.6)$	48	28	63.16%	85.71%
$Cov(\tau = 0.7)$	42	11	79.25%	75.00%
$Cov(\tau = 0.8)$	40	13	75.47%	71.43%
$Cov(\tau = 0.9)$	33	9	78.57%	58.93%

According to the mechanism of our approach, when a relation is mentioned in the given corpus more frequently, the probability that our approach can extract that relation is higher. Thus, by including more documents the recall of our approach increases. We iteratively applied our approach (with the setting $Cov(\tau = 0.7)$ to a NYTimes corpus

of increasing size, starting from 10 years of data up to 21 years. In Figure 4.3, the red line denotes the total number of tagged sentences after our preprocessing step. The total number of included sentences is increased approximately 10,000 in each year. The blue bars show the count of extracted unique company pairs. As the figure shows, by enlarging the size of the dataset, the total amount of extracted targeted relations are mainly increasing. One exception happened when the experimental corpus contains 17 years articles. Our approach extracted only 2,931 unique relations. This number is 41 less than the count of extracted relations based on 16 years articles. That is caused by the total amount of selected patterns is decreased from 15 to 11, since some of the patterns are not representative enough (i.e., with *Cov* scores lower than the threshold) in the larger corpus.

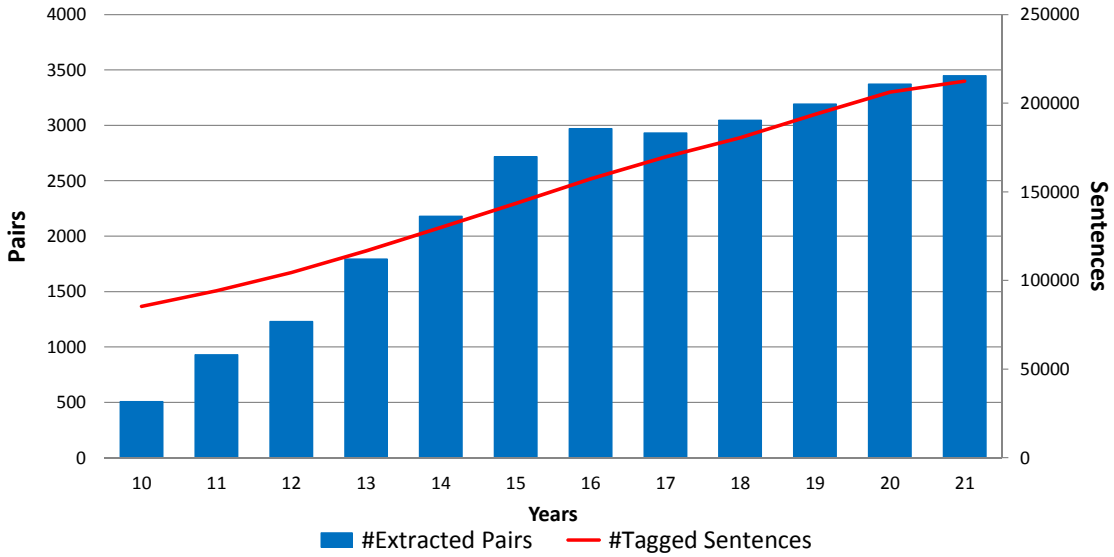


Figure 4.3: The accumulated count of extracted company pairs from subsets of NYTimes corpus

Furthermore, we compared our approach with a state-of-the-art distant learning approach developed by Zeng et al. Zeng et al. (2015). They applied the piecewise convolutional neural networks with multi-instance learning for relation extraction, which we refer as PCNNs. In their experiments, a dataset that contains the New York Times articles labeled with Freebase relations was used[2]. The sentences from 2005 to 2006 were used for training, while the ones from 2007 were used for testing. As we have introduced in Section 4.1, Freebase contains only the major acquisitions of companies, which can be considered as the ownership_of relation. However, all of the instances of the ownership_of relation were mislabeled to be negative in the original dataset. Therefore, to compare PCNNs with our approach for extracting the ownership_of relation, we had to relabel the training set according to the corresponding Freebase relations (99 pairs are matched

[2] http://iesl.cs.umass.edu/riedel/ecml/

in the training set). Since only 14 Freebase relations can be matched in the test set, we randomly picked and manually validated 100 company pairs (including 50 positives and 50 negatives) from the articles in 2007. Table 4.5 shows the evaluation results based on this new test set by applying PCNNs and employing the patterns that were selected by the optimal setting (*Cov* ($\tau = 0.7$)) in our previous experiments. For this specific type of relation, although the precision of PCNNs was extremely high, only 5 pairs were recognized as positive. Our approach, which outperforms PCNNs in both recall and F-measure, can extract more company pairs from the test set with a comparable precision.

Table 4.5: Evaluation comparison (ownership_of relation)

Strategy	**Precision**	**Recall**	**F-measure**
Cov($\tau = 0.7$)	94.7%	36.0%	52.2%
PCNNs	100.0%	10.0%	18.2%

Regarding efficiency, our approach can extract business relations at a rate of about 650 documents per minute on a standard consumer PC, with most of the time spent on preprocessing. The efficiency can be further improved by implementing a distributed system to apply our approach as the strategy introduced by Wang and Min (2015).

Multiple types relations

To evaluate the effect of the holistic pattern identification strategy that was introduced in Section 4.5, we apply our approach to extract the ownership_of and partnership_of relations at the same time. The selection of patterns and new seeds is done using the *Hit* score together with a threshold value of 10. In Table 4.6, we present all patterns that are selected to extract the partnership_of relation by applying the holistic pattern identification strategy. Most of the patterns, especially the top-ranked ones, characterize the partnership_of relation, only several incorrect patterns are included, such as "displaced" and "competitors in".

We also show the selected patterns that are generated by extracting partnership_of relation non-holistically. Without our holistic strategy, the top-ranked patterns mainly represent the ownership_of relation. This problem is caused by a falsely selected pattern (i.e., "owned by"), which leads to more and more patterns that characterize the ownership_of relation. Without the simultaneous extraction of the ownership_of relation to "attract" its patterns, more mistakes are made.

Table 4.6: Key-phrases of selected patterns for partnership_of relation

w/ holistic strategy		w/o holistic strategy	
rank	**Patterns**	**rank**	**Patterns**
1	stake in	1	unit of
2	deal with	2	part of
3	buy	3	parent of
4	invested	4	owned by
5	investment in	5	division of
6	percent of	6	owns
7	agreed to acquire	7	company of
8	group of	8	subsidiary of
9	displaced	9	including
10	partnership with	10	owner of
11	providing	13	percent of
12	made by	14	group of
13	competitors in	15	displaced
14	sign of	18	deal with

CHAPTER 5

Conclusion and Future Work

In this thesis, we focused on the topic of extracting structured information from textual data, which is a typical problem in the domain of information extraction. According to the importance of named entities in text, one of the most important tasks is named entity mining, which is aiming for extracting the relevant information of named entities that are mentioned in text.

In the first chapter, we introduced the motivation of the task of named entity mining, which consists of three major subtasks, named entity recognition, named entity linking, and relation extraction. These three subtasks can build up the entire pipeline of a named entity mining process. By given some textual documents, in the first step, named entity recognition approaches are applied to extract mentions of named entities and identify their types. Then, named entity linking approaches disambiguate mentions of named entities and link them to the corresponding named entities in a knowledge base. Finally, relation extraction approaches focus on extracting relations between extracted named entities.

Chapter 2 introduced state-of-the-art techniques for solving the named entity mining problem. We introduced extracted features, exploited techniques, and benchmark datasets that are frequently used for solving the three tasks separately. Furthermore, except for the three general aspects, we further presented some specific points for each of the tasks as follows: for the named entity recognition task, we discussed various named entity types of interest and the language factor; for the named entity linking task, we described a key component, i.e., knowledge base, in detail; we briefly introduced n-ary relation extraction approaches. Additionally, we discussed the influence of the growth of social media data (especially Twitter data) for named entity recognition and named entity linking tasks.

In Chapter 3, we presented our named entity linking approach, BEL [Zuo et al. (2014)]. The focus of this work has been on lean and light-weight classification algorithms, which

provide a reliable and efficient linking strategy. The comparison of our approach with state-of-the-art techniques on manually-labeled benchmark datasets showed that BEL indeed fulfills the above criteria. Especially on longer, real-world texts, BEL showed an unprecedented quality and efficiency.

In Chapter 4, we introduced our RE approach [Zuo et al. (2017)]. The focus of this work was to extract complex business relations from text efficiently. We are the first to focus on this class of many-to-many relations. To this end, we proposed a relation extraction approach that not only extracts new relations from texts but also indicates their direction in case of non-symmetric ones, such as the ownership_of relation. Another contribution is the holistic pattern identification strategy, which is used to avoid the semantic drift of generated extraction patterns while dealing with multiple business relations simultaneously.

Based on the topics that have been introduced in this thesis, we discuss the future work of our two approaches, as well as the directions that might be followed by further named entity mining systems.

Named entity linking approach

In our experiments, BEL has been proved to be able to produce reliable linking results on general text, such as new articles, and Wikipedia articles. However, further research is needed to understand how such an approach can be optimized for short texts, which typically lack of context and contain highly ambiguous mentions of named entities. Furthermore, in Wikipedia, lots of new articles are included and many modifications of existing articles are submitted as well dynamically. The current version of BEL employs the models that are trained based on a dump of Wikipedia articles. For example, when we apply BEL on the latest news articles, the out-of-date information in the models can make our approach fail to build correct linkages. Therefore, it is important to update the trained model dynamically. Considering the size of Wikipedia, it is expensive to frequently re-train the models. Thus, a solution is to develop a system, which can capture the important changes of Wikipedia contents and update the trained model. Such a system is also useful for augmenting and updating the information contained in knowledge bases.

Relation extraction approach

Considering the further usages based on business relation extraction results, we would like to extract the duration and domain information of relations. The business relations between companies are often complicated. On one hand, relations are changing over time, on the other hand, relations can exist only in some certain domains. For example, in the last ten years, Apple and Samsung became competitors of each other in the domain

of smartphone industry. Therefore, it is interesting to know in which time period and domain two companies participate in a targeted relation. Moreover, as we have discussed, some of the errors in the result of our approach are caused by the misinterpretation of the semantics, e.g., the cancellation of a deal. Thus, we can further improve the performance of our approach by capturing the underlying semantics in text.

Named entity mining in general

As one of the most important topics in information extraction, more approaches will be developed for extracting information of named entities. Further applications will benefit from the development in this field. Several aspects need to be further improved in the future as follows:

Language A lot of work has been done on extracting information from English texts. Nowadays, more and more documents in languages other than English are available and need to be processed. For example, according to the statistics of Wikipedia until December 2017, over 46 million articles are contained in Wikipedia, where only around 11.8% of them are in English. Therefore, it is also important to develop approaches for the texts in other languages or even provide a unified solution (i.e., language-independent approach) for multi-lingual documents.

Specific cases Lots of approaches have been developed to solve the general problem of named entity mining. For instance, the performances of many named entity recognition approaches on recognizing persons, locations, and organizations are usually promising. Systems that can identify other entity types are also interesting for real-world usages. For example, in the biomedical domain, the interested entity types are medicine and disease names. In RE, except for the well-studied relation types, such as the capital city of a country, the birthplace of a person, and the authors of a book, other complex relation types are also of concern. As an example, business relations between companies are beneficial in the domain of business intelligent.

Runtime and Scalability The initial goals of named entity mining task is to automatically extract structured information from large-scale unstructured data. Many state-of-the-art approaches focus on improving the accuracy performance to provide reliable mining results. These improvements are often based on scarifying runtime and scalability by including, for example, complex reasoning processes. However, with the rapid growth of data that need to be processed, not only accuracy but also runtime and scalability of a system are also worth to pay attention to. In particular, the overall runtime of a typical named entity mining system is the sum of that of each subtask, named entity recognition, named entity linking, and relation extraction.

In conclusion, with the rapid growth of available data, it is important to be able to automatically extract structured information from large amount of unstructured data. Based on three major subtasks of named entity mining, one can build a system to extract structured information about named entities hidden in text automatically. The results of each step are valuable for popular domains including business intelligence, knowledge base population, natural language understanding, and many more.

Bibliography

E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the International Conference on Digital Libraries (DL)*, pages 85–94, 2000.

E. Agirre, O. Ansa, E. Hovy, and D. Martínez. Enriching very large ontologies using the WWW. *arXiv preprint cs/0010026*, 2000.

A. Akbik and A. Löser. KRANKEN: N-ary facts in open information extraction. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 52–56, 2012.

E. Alfonseca and S. Manandhar. An unsupervised method for general named entity recognition and automated concept discovery. In *Proceedings of the international conference on general WordNet*, pages 34–43, 2002.

R. Alfred, L. C. Leong, C. K. On, P. Anthony, T. S. Fun, M. N. B. Razali, and M. H. A. Hijazi. A rule-based named-entity recognition for malay articles. In *International Conference on Advanced Data Mining and Applications*, pages 288–299, 2013.

S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. *DBpedia: A nucleus for a web of open data*. The Semantic Web. Springer, 2007.

I. Augenstein, L. Derczynski, and K. Bontcheva. Generalisation in named entity recognition: A quantitative analysis. *arXiv preprint arXiv:1701.02877*, 2017.

D. Aumueller, H.-H. Do, S. Massmann, and E. Rahm. Schema and ontology matching with COMA++. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 906–908, 2005.

A. Bagga and B. Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*, pages 79–85, 1998.

M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction for the web. In *Proceedings of the International Joint Conference on*

Artificial Intelligence (IJCAI), volume 7, pages 2670–2676, 2007.

M. Banko, O. Etzioni, and T. Center. The tradeoffs between open and traditional relation extraction. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*, volume 8, pages 28–36, 2008.

Y. Benajiba, P. Rosso, and J. M. Benedíruiz. Anersys: An Arabic named entity recognition system based on maximum entropy. In *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, pages 143–153, 2007.

A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71, 1996.

I. Bhattacharya and L. Getoor. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):5, 2007.

D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel. Nymble: a high-performance learning name-finder. In *Proceedings of the conference on Applied natural language processing*, pages 194–201, 1997.

D. M. Bikel, R. Schwartz, and R. M. Weischedel. An algorithm that learns what's in a name. *Machine learning*, 34(1-3):211–231, 1999.

C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia-a crystallization point for the Web of Data. *Web Semantics: science, services and agents on the world wide web*, 7(3):154–165, 2009.

W. J. Black, F. Rinaldi, and D. Mowatt. Facile: Description of the ne system used for MUC-7. In *Proceedings of 7th Message Understanding Conference (MUC-7)*, 1998.

K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 1247–1250, 2008.

A. Borthwick. *A maximum entropy approach to named entity recognition*. PhD thesis, New York University, 1999.

L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

S. Brin. Extracting patterns and relations from the world wide web. In *The World Wide Web and Databases*, pages 172–183. Springer, 1999.

R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 9–16, 2006.

R. C. Bunescu and R. J. Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 724–731, 2005a.

R. C. Bunescu and R. J. Mooney. Subsequence kernels for relation extraction. In *Advances in neural information processing systems*, pages 171–178, 2005b.

A. Carlson, J. Betteridge, R. C. Wang, E. R. Hruschka Jr, and T. M. Mitchell. Coupled semi-supervised learning for information extraction. In *Proceedings of the ACM international conference on Web Search and Data Mining*, pages 101–110, 2010.

X. Carreras, L. Màrquez, and L. Padró. A simple named entity extractor using adaboost. In *Proceedings of the conference on Natural language learning at HLT-NAACL*, pages 152–155, 2003.

C.-H. Chang, M. Kayed, M. R. Girgis, and K. F. Shaala. A survey of web information extraction systems. *Knowledge and Data Engineering, IEEE Transactions on*, 18(10): 1411–1428, 2006.

H. L. Chieu and H. T. Ng. Named entity recognition: a maximum entropy approach using global information. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 1–7, 2002.

N. Chinchor and E. Marsh. MUC-7 information extraction task definition. In *Proceeding of the 7th message understanding conference (MUC-7)*, pages 359–367, 1998.

N. Chinchor and P. Robinson. MUC-7 named entity task definition. In *Proceedings of 7th Message Understanding Conference (MUC-7)*, volume 29, 1997.

N. A. Chinchor. Overview of MUC-7/MET-2. Technical report, SCIENCE APPLICATIONS INTERNATIONAL CORP SAN DIEGO CA, 1998.

J. P. Chiu and E. Nichols. Named entity recognition with bidirectional lstm-cnns. *arXiv preprint arXiv:1511.08308*, 2015.

P. Christen. A comparison of personal name matching: Techniques and practical issues. In *ICDM Workshops*, pages 290–294, 2006.

E. Chung, Y.-G. Hwang, and M.-G. Jang. Korean named entity recognition using hmm and cotraining model. In *Proceedings of the sixth international workshop on Information retrieval with Asian languages-Volume 11*, pages 161–167, 2003.

M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in NLP and Very Large Corpora (EMNLP/VLC)*, pages 100–110, 1999.

R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the international*

conference on Machine learning(ICML), pages 160–167, 2008.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

S. Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716, 2007.

M.-C. de Marneffe, B. MacCartney, and C. D. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2006.

G. Demartini, D. E. Difallah, and P. Cudré-Mauroux. ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 469–478, 2012.

G. R. Doddington, A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. Strassel, and R. M. Weischedel. The automatic content extraction (ACE) program-tasks, data, and evaluation. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, volume 2, pages 837–840, 2004.

M. Dredze, P. McNamee, D. Rao, A. Gerber, and T. Finin. Entity disambiguation for knowledge base population. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 277–285, 2010.

F. Du, Y. Chen, and X. Du. Linking entities in unstructured texts with RDF knowledge bases. In *Web Technologies and Applications*, pages 240–251. Springer, 2013.

A. Ekbal and S. Bandyopadhyay. Bengali named entity recognition using support vector machine. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*, pages 51–58, 2008.

R. Elmasri and S. Navathe. Fundamentals of database systems addison wesley. *Reading, MA*, 2003.

M. Elsner, E. Charniak, and M. Johnson. Structured generative models for unsupervised named-entity clustering. In *Proceedings of the conference on Natural language learning at HLT-NAACL*, pages 164–172, 2009.

O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Web-scale information extraction in KnowItAll:(preliminary results). In *Proceedings of the International World Wide Web Conference (WWW)*, pages 100–110, 2004.

O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S.

Weld, and A. Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134, 2005.

A. Fader, S. Soderland, and O. Etzioni. Scaling Wikipedia-based named entity disambiguation to arbitrary web text. In *IJCAI Workshop: User Contributed Knowledge and Artificial Intelligence: An Evolving Synergy*, 2009.

M. Färber, F. Bartscherer, C. Menne, and A. Rettinger. Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web*, pages 1–53, 2016.

D. Farmakiotou, V. Karkaletsis, J. Koutsias, G. Sigletos, C. D. Spyropoulos, and P. Stamatopoulos. Rule-based named entity recognition for greek financial texts. In *Proceedings of the Workshop on Computational lexicography and Multimedia Dictionaries (COMLEX)*, pages 75–78, 2000.

C. Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

T. Finin, W. Murnane, A. Karandikar, N. Keller, J. Martineau, and M. Dredze. Annotating named entities in twitter data with crowdsourcing. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 80–88, 2010.

J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*, pages 363–370, 2005.

M. Fleischman. Automated subcategorization of named entities. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL) (Companion Volume)*, pages 25–30, 2001.

M. Fleischman and E. Hovy. Fine grained classification of named entities. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 1–7, 2002.

M. B. Fleischman and E. Hovy. Multi-document person name resolution. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), Reference Resolution Workshop*, pages 66–82, 2004.

R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. Named entity recognition through classifier combination. In *Proceedings of the Conference on Natural Language Learning at HLT-NAACL - Volume 4*, pages 168–171, 2003.

J. Fukumoto, F. Masui, M. Shimohata, and M. Sasaki. Oki electric industry: Description of the Oki system as used for MUC-7. In *Proceedings of 7th Message Understanding Conference (MUC-7)*, 1998.

J. Gantz and D. Reinsel. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. *IDC iView: IDC Analyze the future*, 2007(2012):

1–16, 2012.

J. Gao, M. Li, A. Wu, and C.-N. Huang. Chinese word segmentation and named entity recognition: A pragmatic approach. *Computational Linguistics*, 31(4):531–574, 2005.

A. Gattani, D. S. Lamba, N. Garera, M. Tiwari, X. Chai, S. Das, S. Subramaniam, A. Rajaraman, V. Harinarayan, and A. Doan. Entity extraction, linking, classification, and tagging for social media: a Wikipedia-based approach. *Proceedings of the VLDB Endowment*, 6(11):1126–1137, 2013.

R. Girju, A. Badulescu, and D. Moldovan. Learning semantic constraints for the automatic discovery of part-whole relations. In *Proceedings of the conference on Natural language learning at HLT-NAACL*, pages 1–8, 2003.

C. H. Gooi and J. Allan. Cross-document coreference on a large scale corpus. In *Proceedings of the conference on Natural language learning at HLT-NAACL*, pages 9–16, 2004.

S. Gottipati and J. Jiang. Linking entities to a knowledge base with query expansion. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 804–813, 2011.

R. Grishman and B. Sundheim. Message understanding conference-6: A brief history. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 466–471, 1996.

T. Gruetze, G. Kasneci, Z. Zuo, and F. Naumann. Coheel: Coherent and efficient named entity linking through random walks. *Web Semantics: Science, Services and Agents on the World Wide Web*, 37:75–89, 2016.

S. Guo, M.-W. Chang, and E. Kiciman. To link or not to link? a study on end-to-end tweet entity linking. In *Proceedings of the conference on Natural language learning at HLT-NAACL*, pages 1020–1030, 2013a.

Y. Guo, B. Qin, T. Liu, and S. Li. Microblog entity linking by leveraging extra posts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 863–868, 2013b.

Z. GuoDong, S. Jian, Z. Jie, and Z. Min. Exploring various knowledge in relation extraction. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*, pages 427–434, 2005.

X. Han and J. Zhao. Named entity disambiguation by leveraging Wikipedia semantic knowledge. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 215–224. ACM, 2009.

D. Hanisch, K. Fundel, H.-T. Mevissen, R. Zimmer, and J. Fluck. Prominer: rule-based protein and gene entity recognition. *BMC bioinformatics*, 6(1):S14, 2005.

M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 539–545, 1992.

J. Hoffart, F. M. Suchanek, K. Berberich, E. Lewis-Kelham, G. de Melo, and G. Weikum. YAGO2: exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 229–232, 2011a.

J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 782–792, 2011b.

J. Hoffart, S. Seufert, D. B. Nguyen, M. Theobald, and G. Weikum. KORE: keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 545–554, 2012.

E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. Ontonotes: the 90% solution. In *Proceedings of the conference on Natural language learning at HLT-NAACL: Short Papers*, pages 57–60, 2006.

H. Isozaki and H. Kazawa. Efficient support vector classifiers for named entity recognition. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 1–7, 2002.

A. Jain, S. Cucerzan, and S. Azzam. Acronym-expansion recognition and ranking on the web. In *Information Reuse and Integration (IRI)*, pages 209–214, 2007.

H. Ji, R. Grishman, H. T. Dang, K. Griffitt, and J. Ellis. Overview of the tac 2010 knowledge base population track. In *Third Text Analysis Conference (TAC)*, volume 3, 2010.

A. Jimeno, E. Jimenez-Ruiz, V. Lee, S. Gaudan, R. Berlanga, and D. Rebholz-Schuhmann. Assessment of disease named entity recognition on a corpus of annotated sentences. In *BMC bioinformatics*, volume 9, page S3. BioMed Central Ltd, 2008.

N. Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*, page 22, 2004.

J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. GENIA corpusa semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl 1):i180–i182, 2003.

J.-D. Kim, T. Ohta, Y. Tsuruoka, Y. Tateisi, and N. Collier. Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications (JNLPBA)*,

pages 70–75, 2004.

S. Kim, K. Toutanova, and H. Yu. Multilingual named entity recognition using parallel data and metadata from wikipedia. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*, pages 694–702, 2012.

Z. Kozareva, B. Bonev, and A. Montoyo. Self-training and co-training applied to Spanish named entity recognition. In *Mexican International Conference on Artificial Intelligence*, pages 770–779, 2005.

S. Lacoste-Julien, K. Palla, A. Davies, G. Kasneci, T. Graepel, and Z. Ghahramani. SiGMa: Simple greedy matching for aligning large knowledge bases. *arXiv preprint arXiv:1207.4525*, 2012.

J. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the international conference on Machine learning (ICML)*, pages 282–289, 2001.

G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

R. Leaman and G. Gonzalez. Banner: an executable survey of advances in biomedical named entity recognition. In *Pacific symposium on biocomputing*, volume 13, pages 652–663, 2008.

J. Lehmann, S. Monahan, L. Nezda, A. Jung, and Y. Shi. Lcc approaches to knowledge base population at tac 2010. In *Text Analysis Conference (TAC)*, 2010.

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

C. Li, J. Weng, Q. He, Y. Yao, A. Datta, A. Sun, and B.-S. Lee. TwiNER: named entity recognition in targeted twitter stream. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 721–730. ACM, 2012.

W. Li and A. McCallum. Rapid development of hindi named entity recognition using conditional random fields and feature induction. *ACM Transactions on Asian Language Information Processing (TALIP)*, 2(3):290–294, 2003.

W. Liao and S. Veeramachaneni. A simple semi-supervised algorithm for named entity recognition. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, pages 58–65, 2009.

G. Limaye, S. Sarawagi, and S. Chakrabarti. Annotating and searching web tables using entities, types and relationships. *Proceedings of the VLDB Endowment*, 3(1-2): 1338–1347, 2010.

T. Lin, Mausam, and O. Etzioni. Entity linking at web scale. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 84–88, 2012.

Y.-F. Lin, T.-H. Tsai, W.-C. Chou, K.-P. Wu, T.-Y. Sung, and W.-L. Hsu. A maximum entropy approach to biomedical named entity recognition. In *Proceedings of the International Workshop on Biological Knowledge Discovery and Data Mining*, pages 56–61, 2004.

X. Liu, S. Zhang, F. Wei, and M. Zhou. Recognizing named entities in tweets. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*, pages 359–367, 2011.

X. Liu, Y. Li, H. Wu, M. Zhou, F. Wei, and Y. Lu. Entity linking for tweets. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*, pages 1304–1311, 2013.

M. Loster, Z. Zuo, F. Naumann, O. Maspfuhl, and D. Thomas. Improving company recognition from unstructured text by using dictionaries. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, pages 610–619, 2017.

F. Mahdisoltani, J. Biega, and F. Suchanek. Yago3: A knowledge base from multilingual Wikipedias. In *Proceedings of the Biennial Conference on Innovative Data Systems Research*, 2014.

G. S. Mann and D. Yarowsky. Unsupervised personal name disambiguation. In *Proceedings of the Conference on Natural Language Learning at HLT-NAACL*, volume 4, pages 33–40, 2003.

D. Maynard, V. Tablan, C. Ursu, H. Cunningham, and Y. Wilks. Named entity recognition from diverse text types. In *Recent Advances in Natural Language Processing (RANLP)*, pages 257–274, 2001.

A. McCallum and W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the conference on Natural language learning at HLT-NAACL*, pages 188–191, 2003.

R. McDonald, F. Pereira, S. Kulick, S. Winters, Y. Jin, and P. White. Simple algorithms for complex relation extraction with applications to biomedical IE. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*, pages 491–498, 2005.

P. McNamee and H. T. Dang. Overview of the tac 2009 knowledge base population track. In *Text Analysis Conference (TAC)*, volume 17, pages 111–113, 2009.

E. Meij, W. Weerkamp, and M. De Rijke. Adding semantics to microblog posts. In

Proceedings of the ACM international conference on Web Search and Data Mining, pages 563–572. ACM, 2012.

P. N. Mendes, J. Daiber, M. Jakob, and C. Bizer. Evaluating dbpedia spotlight for the tac-kbp entity linking task. In *Proceedings of the TACKBP 2011 Workshop*, volume 116, pages 118–120, 2011a.

P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the International Conference on Semantic Systems*, pages 1–8, 2011b.

R. Mihalcea. Using Wikipedia for automatic word sense disambiguation. In *Proceedings of the conference on Natural language learning at HLT-NAACL*, pages 196–203, 2007.

R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242, 2007.

A. Mikheev, C. Grover, and M. Moens. Description of the LTG system used for MUC-7. In *Proceedings of 7th Message Understanding Conference (MUC-7)*, pages 1–12, 1998.

A. Mikheev, M. Moens, and C. Grover. Named entity recognition without gazetteers. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 1–8, 1999.

R. L. Milidiú, J. C. Duarte, and R. Cavalcante. Machine learning algorithms for Portuguese named entity recognition. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 11(36), 2007.

E. Minkov, R. C. Wang, and W. W. Cohen. Extracting personal names from email: Applying named entity recognition to informal text. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 443–450, 2005.

M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th annual meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2*, pages 1003–1011, 2009.

A. Mitchell et al. Annotation guidelines for relation detection and characterization (RDC) Version 3.6, 2002.

S. Monahan, J. Lehmann, T. Nyberg, J. Plymale, and A. Jung. Cross-lingual cross-document coreference with entity linking. In *Text Analysis Conference (TAC)*, 2011.

A. Moro and R. Navigli. Integrating syntactic and semantic analysis into the open information extraction paradigm. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2148–2154, 2013.

D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

D. Nadeau, P. D. Turney, and S. Matwin. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 266–277, 2006.

F. Naumann and M. Herschel. *An Introduction to Duplicate Detection.* Morgan and Claypool Publishers, 2010.

G. Navarro. A guided tour to approximate string matching. *ACM computing surveys (CSUR)*, 33(1):31–88, 2001.

J. Nothman, N. Ringland, W. Radford, T. Murphy, and J. R. Curran. Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence*, 194:151 – 175, 2013.

G. Paliouras, V. Karkaletsis, G. Petasis, and C. D. Spyropoulos. Learning decision trees for named-entity recognition and classification. In *ECAI Workshop on Machine Learning for Information Extraction*, 2000.

M. Pasca, D. Lin, J. Bigham, A. Lifchits, and A. Jain. Organizing and searching the world wide web of facts-step one: the one-million fact extraction challenge. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, volume 6, pages 1400–1405, 2006.

T. Pedersen, A. Purandare, and A. Kulkarni. Name discrimination by clustering similar contexts. In *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, volume 3406 of *Lecture Notes in Computer Science*, pages 226–237. Springer, 2005.

T. Pellissier Tanon, D. Vrandečić, S. Schaffert, T. Steiner, and L. Pintscher. From Freebase to Wikidata: The great migration. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 1419–1428, 2016.

N. Peng, H. Poon, C. Quirk, K. Toutanova, and W.-t. Yih. Cross-sentence n-ary relation extraction with graph lstms. *Transactions of the Association for Computational Linguistics*, 5:101–115, 2017.

G. Petasis, F. Vichot, F. Wolinski, G. Paliouras, V. Karkaletsis, and C. D. Spyropoulos. Using machine learning to maintain rule-based named-entity recognition and classification systems. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*, pages 426–433, 2001.

D. Ploch, L. Hennig, E. W. De Luca, S. Albayrak, and T. DAI-Labor. Dai approaches to the TAC-KBP 2011 entity linking task. In *Text Analysis Conference (TAC)*, 2011.

D. Ploch, L. Hennig, A. Duka, E. W. De Luca, and S. Albayrak. Gerned: A german cor-

pus for named entity disambiguation. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 3886–3893, 2012.

L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

K. Riaz. Rule-based named entity recognition in urdu. In *Proceedings of the named entities workshop*, pages 126–135, 2010.

E. Riloff, R. Jones, et al. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 474–479, 1999.

A. Ritter, S. Clark, O. Etzioni, et al. Named entity recognition in Tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1524–1534, 2011.

J. Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61: 85–117, 2015.

B. Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 104–107, 2004.

W. Shen, J. Wang, P. Luo, and M. Wang. LINDEN: linking named entities with knowledge base via semantic knowledge. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 449–458, 2012.

W. Shen, J. Wang, and J. Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460, 2015.

A. Sil, E. Cronin, P. Nie, Y. Yang, A.-M. Popescu, and A. Yates. Linking named entities to any database. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 116–127, 2012.

R. Snow, D. Jurafsky, and A. Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In *Advances in neural information processing systems*, pages 1297–1304, 2005.

S. Soderland. Learning to extract text-based information from the World Wide Web. In *Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD)*, volume 97, pages 251–254, 1997.

S. Strassel, A. Mitchell, and S. Huang. Multilingual resources for entity extraction. In *Proceedings of the ACL workshop on Multilingual and mixed-language named entity recognition-Volume 15*, pages 49–56, 2003.

F. M. Suchanek, G. Kasneci, and G. Weikum. YAGO: a core of semantic knowledge. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 697–706, 2007.

F. M. Suchanek, J. Hoffart, E. Kuzey, and E. Lewis-Kelham. Yago2s: Modular high-quality information extraction with an application to flight planning. In *Proceedings of the Conference Datenbanksysteme in Business, Technologie und Web Technik (BTW)*, volume 214, pages 515–518, 2013.

E. F. Tjong Kim Sang and S. Buchholz. Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning-Volume 7*, pages 127–132, 2000.

E. F. Tjong Kim Sang and F. De Meulder. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In *Proceedings of the conference on Natural language learning at HLT-NAACL*, volume 4, pages 142–147, 2003.

D. Vrandečić and M. Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

C. Walker, S. Strassel, J. Medero, and K. Maeda. ACE 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57, 2006.

T. Wang and H. Min. Entity relation mining in large-scale data. In *Database Systems for Advanced Applications: DASFAA 2015 International Workshops, SeCoP, BDMS, and Posters*, page 109, 2015.

M. Weis and F. Naumann. Dogmatix tracks down duplicates in XML. In *Proceedings of the ACM SIGMOD international conference on Management of data*, pages 431–442. ACM, 2005.

I. H. Witten and D. N. Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceedings of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy*, pages 25–30, 2008.

F. Wu and D. S. Weld. Automatically refining the Wikipedia infobox ontology. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 635–644, 2008.

F. Wu and D. S. Weld. Open information extraction using Wikipedia. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*, pages 118–127, 2010.

Y. Yang and M.-W. Chang. S-mart: Novel tree-based structured learning algorithms applied to tweet entity linking. *arXiv preprint arXiv:1609.08075*, 2016.

D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. *The*

Journal of Machine Learning Research, 3:1083–1106, 2003.

D. Zeng, K. Liu, S. Lai, G. Zhou, J. Zhao, et al. Relation classification via convolutional deep neural network. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 2335–2344, 2014.

D. Zeng, K. Liu, Y. Chen, and J. Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1753–1762, 2015.

C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, 22(2):179–214, Apr. 2004.

C. Zhang. *DeepDive: a data management system for automatic knowledge base construction.* PhD thesis, The University of Wisconsin-Madison, 2015.

T. Zhang and D. Johnson. A robust risk minimization based named entity recognition system. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 204–207, 2003.

S. Zhao and R. Grishman. Extracting relations with integrated information using kernel methods. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*, pages 419–426, 2005.

Z. Zheng, F. Li, M. Huang, and X. Zhu. Learning to link entities with knowledge base. In *Proceedings of the conference on Natural language learning at HLT-NAACL*, pages 483–491, 2010.

G. Zhou and J. Su. Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*, pages 473–480, 2002.

J. Zhu, Z. Nie, X. Liu, B. Zhang, and J.-R. Wen. StatSnowball: a statistical approach to extracting entity relationships. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 101–110, 2009.

Z. Zuo, G. Kasneci, T. Gruetze, and F. Naumann. BEL: Bagging for entity linking. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 2075–2086, 2014.

Z. Zuo, M. Loster, R. Krestel, and F. Naumann. Uncovering business relationships: Context-sensitive relationship extraction for difficult relationship types. In *Proceedings of the Conference "Lernen, Wissen, Daten, Analysen" (LWDA)*, 2017.

Selbstständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Doktorarbeit mit dem Thema:

From unstructured to structured:
Context-based Named Entity Mining from Text

selbststandig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Potsdam, den 5. Dezember 2017

Zhe Zuo