Unsupervised and Supervised Learning of Complex Relation Instances Extraction in Natural Language

Version of November 13, 2020

Zina Wang

Unsupervised and Supervised Learning of Complex Relation Instances Extraction in Natural Language

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Zina Wang
born in Hunan, China

Cyber Security Research Group
Department of Computer Science
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
www.ewi.tudelft.nl

ZyLAB Technologies BV
Laarderhoogtweg 25
Amsterdam, the Netherlands
https://zylab.com/

Unsupervised and Supervised Learning of Complex Relation Instances Extraction in Natural Language

Author: Zina Wang
Student id: 4716213

Abstract

Relation extraction has been considered as one of the most popular topics nowadays, thanks for its common application in knowledge graph, machine reading and other artificial intelligence sub-field. However, this field has long been suffered from data hunger. Annotating large high-quality datasets for relation extraction is troublesome and time-consuming. This thesis project will main focus on efficient way of annotating text datasets for extracting complex relations between entities. Moreover, we put some efforts on compare the influence of different components in the pipeline. The main contributions of this project are the comparisons and analysis regarding the influences of components, which are in place for the majority of relation extraction models, and the clear literature review together with the summary of available datasets in the relation extraction flow.

Thesis Committee:

Chair:	Prof.dr.ir. Inald Lagendijk, Faculty EEMCS, TU Delft
University supervisor:	Dr. Sicco Verwer, Faculty EEMCS, TU Delft
Company supervisor:	Johannes C. Scholtes, Zylab

Preface

Zina Wang
Delft, the Netherlands
November 13, 2020

Contents

Preface iii

Contents v

List of Figures vii

1 Introduction 1
- 1.1 Background . . . 1
- 1.2 Problem Statement . . . 4
- 1.3 Research Question . . . 5
- 1.4 Contribution . . . 6

2 Related Work 9
- 2.1 Automatic Annotation Approaches for Relation Extraction . . . 9
- 2.2 Mainstream Approaches of Detecting Relations . . . 14
- 2.3 Gap Investigation . . . 16

3 Methodology 19
- 3.1 Data Collection . . . 19
- 3.2 Benchmark Selection . . . 26
- 3.3 Model Construction . . . 32

4 Experiment 41
- 4.1 Data Preprocessing and Analysis . . . 42
- 4.2 Evaluation Metric . . . 46
- 4.3 Result Analysis . . . 46

5 Discussion 59
- 5.1 Conclusion . . . 59
- 5.2 Future work . . . 60

Bibliography **61**

List of Figures

1.1 The Overview of Concepts 2
1.2 An example of entity linking[86] 3

2.1 The Overview of Related Work of Annotation Approaches 14
2.2 The Overview of Related Work of Relation Extraction Models 16

3.1 The Overview of the Methodology Chapter 19
3.2 The Distribution of Used Datasets 20
3.3 An Example of Overlapping Relations[107] 28
3.4 The Structure of Benchmark Model 30
3.5 The Structure of the Encoder and the Decoder[107] 31
3.6 An Example of Data in Freebase[24] 32
3.7 Experiment Pipeline 33
3.8 Experiment Design Part 1 34
3.9 The Framework of Flair NER[6] 35
3.10 The Visualization of the Difficulty of Entity Linking Tasks[27] 36
3.11 The pipelien of DoSer for Entity Disambiguation Tasks[113] 37
3.12 Experiment Design Part 2 39
3.13 The Framework of Flair Embeddings[5] 40

4.1 The Distribution of Tokens in Training Set 2 45
4.2 The Distribution of Tokens in Test Set 2 45
4.3 The Visualisation of Complicated Relation Instances 46
4.4 The Performance of Original Relation Extraction Model in Validation Set 2 . . 48
4.5 The Performance in Validation Set of Relation Extraction Model with NLTK NER 50
4.6 The Performance in Validation Set of the Relation Extraction Model with Flair NER 50
4.7 The Visualisation of Different Maximum Length of Documents in Training set . 51
4.8 The Performance in Validation Set of the Relation Extraction Model with Max 85 words in Training set 52

4.9 The Performance in Validation Set of Relation Extraction Model with Max 70 words in Training set . . . 52
4.10 The Performance in Validation Set of the Relation Extraction Model with Max 55 words in Training set . . . 53
4.11 The Performance in Validation Set of the Relation Extraction Model with Flair NER and Max 85 words in Training Set . . . 54
4.12 The Performance in Validation Set of the Relation Extraction Model with Flair NER and Max 70 words in Training Set . . . 55
4.13 The Performance in Validation Set of the Relation Extraction Model with Flair NER and Max 55 words in Training Set . . . 55
4.14 The F1-Score in Test Set with the Combination of Different NERs and Different Max Length of Sentence in Training set . . . 56
4.15 Recall in Test Set with the Combination of Different NERs and Different Max Length of Sentence in Training set . . . 57
4.16 Precision in Test Set with the Combination of Different NERs and Different Max Length of Sentence in Training set . . . 57

Chapter 1

Introduction

In this chapter we will cover the concepts and background knowledge that form the foundation of this thesis. First of all, we will specify terminology definitions followed by an explanation of our problem statement. The chapter will conclude with our research question and contribution. After reading this chapter, readers should have an overall impression of the field and be familiar with our research goals.

1.1 Background

Intelligence is known as the ability of understanding logic, comprehending complex concepts and learning knowledge through experience and the acquisition of information. Intelligence was long thought to be an ability reserved for humans and animals. However, due to the advancement of computer technology, Artificial Intelligence (AI) has emerged as a new category of intelligence. AI has been gaining considerable attention from researchers in numerous fields, such as mechanics, vehicles, vision and media, in which it serves as a tool to improve performance and efficiency. AI consists of various subfields including machine learning. In order to provide a clear scope of our research, we have visualized the hierarchy of the subfields within AI in Figure 1.1. As the figure shows, this project will be focusing on Knowledge Base Population (KBP) and its subset of fields.

As one of the most popular fields of AI, Machine learning was coined by Samuel [83]. Machine learning is referred as computer programs with the capability of learning from experience, similar to how human brains function. Traditional machine learning techniques use statistical models and algorithms for classification and prediction tasks. Through pattern recognition, machine learning models are able to mimic how human brains classify objects and make decisions. Recent developments in the field of machine learning predominantly focus on neural networks, the concept of which was inspired by biological neural networks[19]. Compared to traditional manual feature engineering work in machine learning algorithms, neural-based models contain multiple layers of nonlinear information processing to extract features automatically for classification and prediction tasks. Addressed by Bengio et al. [10] and Schmidhuber [84], the learning process of AI models can be supervised, semi-supervised or unsupervised. In supervised learning, the model learns features

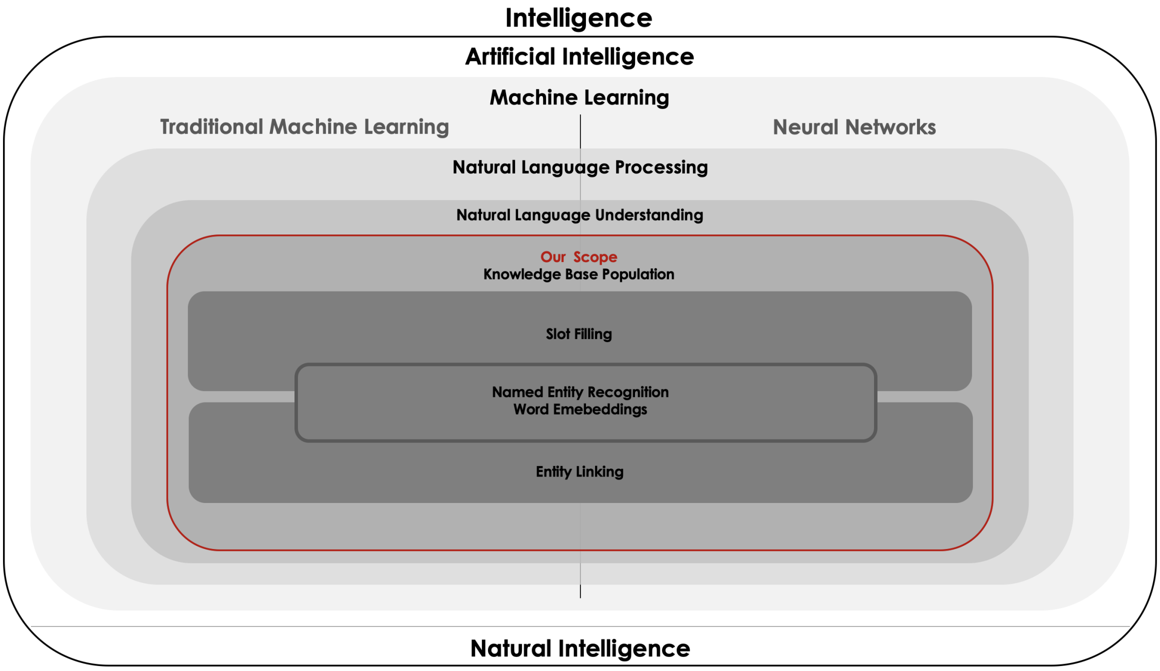


Figure 1.1: The Overview of Concepts

from data with its desired output values or labels. Learning partially with labels or without labels entirely is called semi-supervised learning and unsupervised learning respectively.

As a subfield of machine learning, Natural Language Processing (NLP) is a research topic which focuses on the interactions between computer languages and human languages. Based on probability and dependency between words, NLP tasks aims to process, understand and even generate natural languages. One of the main subfields of NLP is Natural Language Understanding (NLU), which mainly deals with computer comprehension of natural languages. Through NLU, the computer can read, write, collect and analyze texts. Hence, it is fundamental to a number of other fields such as machine translation [53], question answering [34, 102] and document classification.

Knowledge base population (KBP) is used for describing systems which aim to extract valuable information from unstructured natural languages with the goal of populating an emergent knowledge base[29]. One typical example of KBP is filling in incomplete information in Wikipedia Infobox using Wikipedia. For humans, a knowledge base can have the form of experience, common sense and knowledge learned. Machine-readable popular knowledge bases (KBs) include Wikipedia, YAGO [88], Freebase [11] and DBpedia [8], as mentioned by Shen et al. [86].

Slot filling and entity linking are two subtasks of KBP. The goal of slot filling is to collect certain information about a specific entity. For example, for *Person* Bob who is mentioned in a paragraph, answering questions like where he graduated or who his friends using a knowledge base is a slot filling task. And if we have target relations ”graduate from”

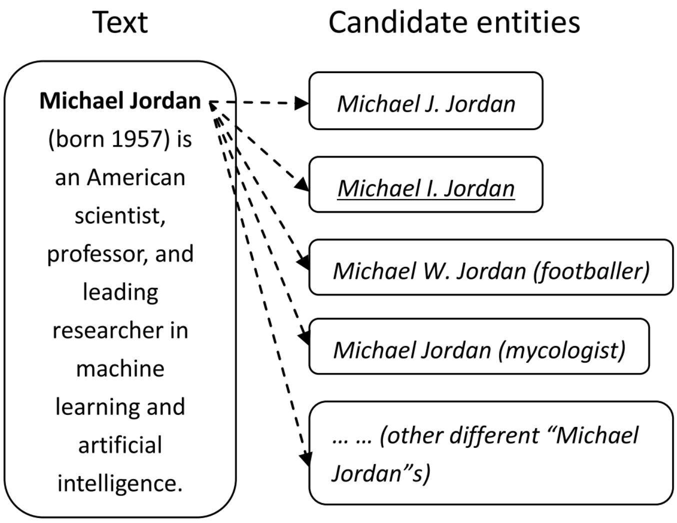

Text
Candidate entities
Michael Jordan
(born 1957) is an American scientist, professor, and leading researcher in machine learning and artificial intelligence.
Michael J. Jordan
Michael I. Jordan
Michael W. Jordan (footballer)
Michael Jordan (mycologist)
… … (other different "Michael Jordan"s)

Figure 1.2: An example of entity linking[86]

and "founder of", then slot filling is the task for finding corresponding entities to fill in the slots in the triplets $(em_i, em_j, graduate\_from)$ and $(em_m, em_n, founder\_of)$. In other words, the goal of slot filling is to find corresponding entities into pre-defined relation slots.

In the entity linking task, variations of an entity mention are taken and linked with concrete entities in the knowledge base. The main task that entity linking aims to perform is to find which of the entities in the sentence is the subject. This ambiguity issue is caused by text variations and ambiguities in the sentence. For instance, Michael Jordan might be referred to the basketball player Michael Jordan or a mycologist Michael Jordan. The goal of entity linking is to clarify which entity mentions we are talking about in order to reduce text ambiguities. An example can be found in Figure 1.2.

The common technique between slot filling and entity linking is relation extraction (RE) [82]. As a subtask of information extraction (IE), RE is the task of extracting relations among entities from unstructured text. By converting words from natural languages into mathematical vectors, the computer is able to interpret the semantic meanings of natural languages. These mathematical vectors, as serve as representations of words, are called word embeddings. Word embeddings allows words to be represented in a continuous and multidimensional vector space. This makes it possible to easily compare the semantic meaning of two words, as it is based on their vector distance. Introduced by [55], Word2Vec uses shallow neural networks to learn word embeddings. Other implementations include global vectors (GloVe) [65], Bert [20] and Flair [2].

To provide some more insights into KBP, we can consider the following examples. Assuming we have a *sentence* s: 'Stephen William Hawking was the Lucasian Professor of Mathematics at the University of Cambridge between 1979 and 2009', with 18 *words* $s = [w_1, w_2, w_3, ..., w_{18}]$. The subset of words (w_1, w_2, w_3) (Stephen William Hawkin) can be detected as a person *entity mention* em_1. Another subset of words (w_6, w_7, w_8) (Lucasian

Professor of Mathematics) can be detected as a job position *entity mention* em_2. If the *relation mention* rm_1, in this case 'job_title', can be found between em_1 and em_2, it is fair to state that *triplet* $t_1 = [em_1, em_2, rm_1]$ exists in the *sentence* s. The process of detecting entities, such us person name Stephen William Hawkin, is called Named Entity Recognition (NER), which is one of the most important techniques in NLP.

1.2 Problem Statement

Unlike decades ago, nowadays, people are overwhelmed by a constant flow of information coming from various channels such as television, emails, messages and social media. As the data-driven world is proliferating, traditional information retrieval techniques can no longer fulfill present-day requirements. The need of processing the current ocean of information in an efficient manner leads to the great demand of automatic information extraction. Relation extraction, a subfield in information extraction, is one of the most popular topics in the field of KBP. This conclusion can be proved by the fact that from 2009, *Text Analysis Conference(TAC)* organizes the KBP Track per year for researchers and companies and it has the increasing number of participants, which shows the increasing interest and demand that is put into this field. Even though significant progress has been by researchers in the field of KBP, there are a number of challenges that are yet to be solved. Three of them, which are described below, catch our eyes especially.

Challenge 1: Expensive Annotation

Firstly, it is time-consuming and labor-intensive to manually create and maintain corpora for relation extraction. The high expense of labelling and the limited applications of biased models lead to the high demand of extracting relations automatically and artificially. For example, for the creation of an annotated dataset for the large corpus from Stanford University it takes around $0.13 per annotation and $160,000 in total[7]. If we are able to extract relations without labels or with less manual efforts, the benefits of this will be propagated to many fields such as NLU, knowledge graph construction and search engines. A common solution for this problem is the crowd-sourcing technique, in which a group of crowd workers is employed to annotate the corpus. However, it is not feasible to apply this kind of solutions in every case in all industries. Extracting structured data from unstructured information automatically will significantly contribute in helping people retrieve information efficiently.

Challenge 2: Text Variants and Ambiguities

Secondly, a general model may need large corpora for training and learning. A high amount of data introduces a high amount of noise and confusion. On the one hand, different ways of conveying the same information exist in different texts. For instance, we may refer to a person or location in different manners.We can call the king of Netherlands directly as 'The King' or 'Willem-Alexander'. When the whole paragraph is talking about Netherlands, will the algorithm always understand that 'The King' is referring to The King of Netherlands instead of the historical drama film based on several plays from William Shakespeare's "Henriad"? Being surrounded by noise and variants of expressions makes extraction of key information more challenging. On the other hand, pronouns are very commonly used in natural languages. When there are more than three people or locations in texts, it will be confusing that 'he' and 'it' are referring to whom and what. Even if we use powerful word embeddings to express semantic meaning, AI models may get lost in complex grammatical expressions and semantic structure. As humans we are able to consider this contextual information to understand natural language, but for computer this is one of the biggest obstacles for successfully interpreting text.

Challenge 3: Error Propagation

Thirdly, NLU tasks are troublesome to be solved by using a single model. Simply put, children learn a language in several stages, starting from remembering the alphabet, followed by learning vocabulary and grammar. Limited vocabularies can result in incomplete or incorrect of content. As one can imagine, it is an impossible task to create complex sentence structures without competent language skills. The same is the case for computers. AI models need guidance in cleaning text and detecting candidates of valuable information which can be used for further complicated tasks. A simple example can be that when there is a typo in our conversation: Steve Jobs is an entrepraneur. Human can easily auto-correct in our mind according to the experience. However this kind of mistakes will result in missing of being detected as an noun, therefore missing of being recognized as an mention entity. Error in these prerequisites has great influences on the performance of machine comprehension tasks.

1.3 Research Question

Considering the aforementioned challenges, in this thesis, we aim to build a pipeline for the extraction complex relations from texts with the least manual efforts. The observed challenges inspire the experimental design. Firstly, due to the high costs of annotations, in literature review we tend to prefer the unsupervised annotation models, with the detailed comparisons with supervised and/or semi-supervised models. This means researches, which focus on supervised methods and only provide comparisons between supervised models, are less interesting to this thesis.

Secondly, Named Entity Recognition and word embeddings, Entity Linking are the key language modeling and feature learning for machine to understand natural language. By projecting words into high dimensions and representing them as vectors, powerful word embedding algorithms are targeting to understand texts, text variants and ambiguities and recognizing the minor meaning similarity and/or difference between words as human being does. Named Entity Recognition is capable of filtering and labelling relation instance candidates for relation extraction. It provides labels like *Person, Location* and *Organization* to reduce the confusions in Natural Language Understanding. Entity Linking aims at linking the target in text with the correct entity in the Knowledge Base for clarification purposes. So

Their ability of addressing text variants and ambiguities is the reason why we spend efforts on investigating different word embeddings and Named Entity Recognition.

Last but not least, we introduce the maximum length of documents as an experimental aspect, considering the fact that document length is highly relevant to complexity of contexts and relations. We make the decision of conducting cross experiments between different components of the pipeline for the reason that this will provide insights of how error propagate inside of the pipeline. We believe knowing what exactly goes wrong can navigate us to the fastest way to make it right.

To narrow down the scope of our research, we will mainly focus on the task of relation extraction, which is the key technique in the topic of NLU. In order to meet the needs of large organizations and projects, complex relations are defined as relations between people and locations, people and organizations and people among each other. The main research question and several subquestions are specified below.

- **The Main Question:**
  How to improve the performance of relation extraction?

- **Subquestion 1:**
  Except the model itself, what are influential components in a relation extraction pipeline?

- **Subquestion 2:**
  How can these components affect the performance of relation extraction?

1.4 Contribution

To address the challenge of expensive annotation, we have done comprehensive literature review on current intelligent annotation models and available datasets, which provides insights of possible solutions for researches regarding the annotation topic. On the top of it, we proposed a pipeline of complex relation extraction from unlabeled plain text based on distant supervision and sequence-to-sequence learning with copy mechanism. We selected four aspects to dig deeper, namely Named Entity Recognition, maximum length of documents in dataset, word embeddings and Entity Linking. The selections of these aspects are supported by careful analysis. The combinations of variables from these aspects and corresponding influence on the performance of relation extraction model are analyzed.

Supported by our experiment results, investigating the insights of the model instead of changing the framework can dramatically improve the performance of relation extraction. This conveys the message that the framework of the model is not the only decisive factor. The dependencies and insights of components worth more attention.

In details, the Named Entity Recognition model, which works best in identifying relation instances in documents with longer sentences, can improve the performances of relation extraction by 8.9% in our case. It is also proved that proper combinations of components and configurations, which fits the preferences of components in the relation extraction, can benefit the result (4.1% improve of F1-Score of the general worst performing model in our experiment). Additionally, promising test results can be achieved though we train the model in a smaller and simpler training set. This could potentially save significant amount of time for training models.

To our knowledge, current researches concentrate on new frameworks to solve challenges in the fields but barely investigate how influential the shared common components are. We hereby address the gap. Hopefully, these results will be able to guide future researches to improve the performances of the relation extraction models effectively. Our mindset is that models work well for reasons, vice versa. Understanding insights of models will provide the necessary help to improve models and will extend the application areas of research work in industries.

Chapter 2

Related Work

This project will focus on extracting structural data, such as the infobox in a Wikipedia page, from texts with the least amount of labels. Various researchers have put forward different ideas to label complicated relations and detect them in texts. What we are looking for is an efficient way to annotate high-quality datasets for relation extraction and an accurate model to detect or predict relations in the annotated dataset. Intuitively, supervised models provide better results than unsupervised models as they are trained by labelled data. In reality, we are unable to collect perfect datasets for different research topics. Consequently, this literature review aims to answer the following questions:

- What options do we have to annotate datasets for relation extraction automatically?
- Which annotation approach is the most common used one and why?
- What are mainstream approaches of relation extraction?
- What are their limitations and strengths?
- What is the gap between existing methods and our expectations?

The first two questions will be covered in the section of **Automatic Annotation Approaches for Relation Extraction** and the third and forth questions will be answered in details in the section of **Mainstream Approaches of Detecting Relations**. Finally, we will present an overview and the gap we have identified in the **Gap Investigation** section.

2.1 Automatic Annotation Approaches for Relation Extraction

In this section, we will illustrate existing annotation methods for relation extraction. For clarification, we categorize researchers into several groups, namely distant supervision, pattern-based approaches and other approaches. Also, we make a summary at the end to address the key observations.

2.1.1 Distant Supervision

Distant supervision is one of the most popular approaches for annotating datasets of relation extraction. The earliest system with the spirit of distant supervision dates back decades[12], and is called *DIPRE*. *DIPRE* extracts *date-of-birth* information from web pages for *Mozart*. Originally discussed by Mintz et al. [57], distant supervision is based on the idea that any sentence that contains a pair of entities that appearing a pre-trained knowledge base relation is more likely to be a relevant instance of a corresponding relation. Popular examples of knowledge bases (KBs) include Wikipedia Infoboxes, YAGO[88], Freebase[11] and DBpedia[8]. Moreover, due to the broad knowledge bases, distant supervision does not suffer from overfitting and domain dependence. Consequently, it may be used in different fields and domains. Additionally, distant supervision is able to generate canonical names for relations instead of following defined patterns. It clusters relations into several groups and then maps these groups into relation types.

Distant supervision is an unsupervised approach that can be applied to text with data from knowledge bases without labels[75]. Strictly speaking, distant supervision, an unsupervised approach, can be used in supervised methods as a choice of annotation methods. Some researchers leverage distant supervision as a technique to extract relations directly. While others apply distant supervision as a way of annotating datasets before they extract features for supervised models[90]. Another notable point is that the assumption of distant supervision involves that a pair of entities is able to have a single relation. According to our experience, relations vary when they appear in different situations. Thus, many researchers are attempting to alleviate distant supervision of this hard assumption [75, 35]. Moreover, finding a way of selecting correct instances for interesting relations is also a popular topic of research. Ji et al. [36] proposed a sentence-level attention model to select the instances from knowledge bases, which makes supervision information more accurate and reliable. Furthermore, distant supervision is also based on the assumption that our knowledge base is complete[7].

The aforementioned limitations cannot be ignored. As a result of the strict assumption of distant supervision, our annotations will be noisy as the co-occurrence of two entities is not an indicator of a specific relation type. Based on this assumption, this means we will have many false positives. In the 2014 KBP English Slot Filling, only 16% of a sample of positive training instances expressed the target relation the authors were looking for, as mentioned by Soderland et al. [87]. This is the reason that a significant amount of research is dedicated to reducing noise and decreasing false positives of distant supervision to improve its performance. For example, Phi et al. [70] have combined ranking algorithms with distant supervision to remove the noise. Moreover, Qin et al. [72] has explored a deep reinforcement learning strategy to generate a false positive indicator allowing us to remove them and to avoid error propagation.

2.1.2 Pattern-based approaches

Bootstrapping

Bootstrapping, also called seed selection, is a way to annotate text based on selected seeds. Imagining that there are a container with various types of stones, we repeat this process as many times as desired: randomly pick the stone from the container and record its features as numbers. Then we replace the stone with the numbers we registered. At some point, there is a set of numbers, which is called the selected seed, in the container and the rest are stones, which are unlabelled data. Making use of the recorded numbers or vectors, we populate this representatives into the whole container by assuming that statistical inference can be drawn by the selected stones to estimate the distribution characteristics of all the stones in the container. With this, we are able to populate our annotated datasets by detecting similar seeds and patterns of selected seeds in the first phase.

Bootstrapping helps us label datasets iteratively by adding new seeds and patterns based on the original seeds. Besides, it is a minimally supervised method to find similar instances, as it is based on a small part of labeled datasets[70]. In the KBP 2014 English Slot Filling Track competition, *Beijing University of Posts and Telecommunications* applied a bootstrapping method based on dependency tree paths and achieved above-average results[89]. In order to reduce manual work of selecting seeds in the first stage, Eisner and Karakos [21] applied an approach to bootstrap the system to rank many candidate seeds automatically. Similarly, Kozareva and Hovy [43] proposed a regression model to evaluate the quality of each seed so that following tasks can start from high-quality seeds. Moreover, Phi et al. [70] and Kiso et al. [41] have used the HITS algorithm to rank seeds based on the Espresso algorithm[64].

Because the idea of bootstrapping is to find instances which are similar to initial seeds and patterns. These methods of ranking seeds help models filter informative seeds and patterns. However, the problem is that bootstrapping has always proved to have low precision in researches like Ravichandran and Hovy [73], Mintz et al. [57]. They also mentioned that semantic drift, which is the case when the understood meaning of some words are radically different from the truth or original meaning, is commonly shown in bootstrapping. This is for the reason that there are numerous forms of expressing a specific relation[103]. Moreover, error propagation is one of the common issues of bootstrapping. This is a consequence of the assumption that the seeds are complete and representative enough which is not the case in reality. Intuitively, by random sampling, it is possible to pick the same stone more than once from the container and the sampling size is generally small considering the size of the dataset.

Open Information Extraction

Open Information extraction (IE) is an unsupervised pattern-based methods that aims to realize relation extraction. It was first introduced by TextRunner[9]. Currently, publicly available programs like ReVerb[23], OLLIE[85] and ClausIE[18] are able to annotate binary relations of entities. This approach is characterized by that it is not for finding predefined relation types but for unbounded relations[42]. Open relation extraction focuses on lexicosyntactic patterns, such as *A Verb B*. This approach works ”out-of-the-box”, which means we do not need to prepare a training phase for new domains[87]. However, after evaluating ReVerb, OLLIE and ClausIE with a test dataset, we notice the existence of some obvious downsides to this approach. Open IE only detects those relations in sentences where an explicit relation phrase can be found. For instance, Open IE is unable to find a job title when the exact job title is not included as a verb in the sentence. In this particular example, it is difficult for Open IE to extract *job: title* from a sentence such as ”Dutch journalist Gideon Levy reported...”. Although *journalist* is obvious enough to define as a job title, Open IE is only able to find ”reported”. Moreover, Open IE tends to detect duplicate phases for one relation type. For example, for a sentence like ”Emma Watson is an English actress who was studying at Brown University and Worcester College, Oxford”, Open IE may return ”Emma Watson, is, an English actress”, ”Emma Watson, is, an English actress who was studying” and ”Emma Watson, is, an English actress who was studying at Brown University and Worcester College, Oxford”. These results provide a considerable amount of duplicate information, and a cleaning process is still needed to generate the final annotated datasets. As the method is based on a set of patterns, limited patterns result in fundamental limitations in recall. Researchers like Soderland et al. [87] work on adding relation-specific rules to improve recall.

2.1.3 Other approaches

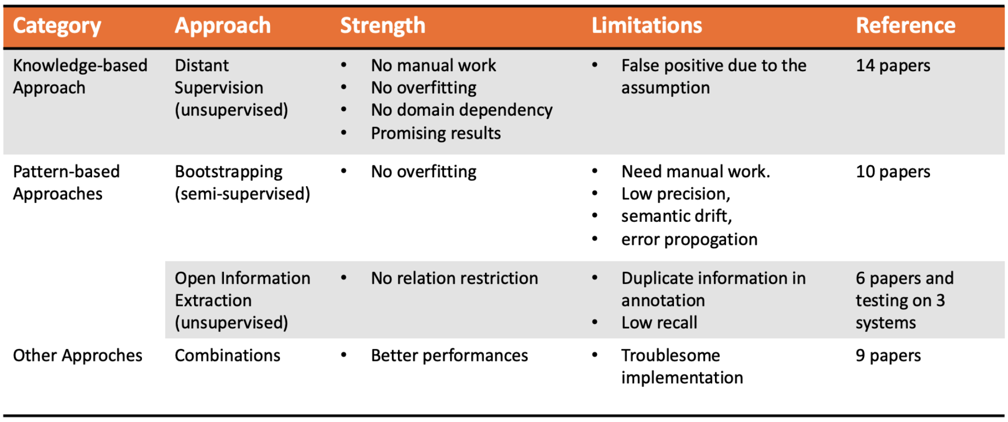
Other methods can be combinations of the aforementioned popular approaches. For example, Angeli et al. [7] proposed three criteria for selecting examples to annotate. Their work can be seen as a combination of distant supervision and bootstrapping, as they attempt to utilize the concept of seed selection to pick up perfectly annotated examples. The paper combines perfect examples with labeled data supported by distant supervision. From their experiments, their model yields 3.9% increase in the 2013 KBP Slot Filling. Generally, these methods are using concepts from various other approaches in order to combine their particular advantages. Nonetheless, it is difficult to avoid disadvantages while combining various methods. Besides, none of the approaches can blindly apply their models to other languages, and these approaches are focusing on details of sentences instead of global structures of all texts[103]. The latter observation will bring valuable missing information. Additionally, Soderland et al. [87] combine Open IE and distant supervision (MULTIR system)[35] together. [103] proposed an effective unsupervised method based on graph mining and *PageRank* algorithm. They constructed extended knowledge graphs for each sentence. Then, all candidates were ranked within their graph to find triggers, which were defined as the smallest extent of a text which most clearly indicates a slot type. Following this, the corresponding relations using these triggers were located. Over state-of-the-art English slot filling approaches, an improvement of 11.6%-25% for the F1-score for different relation types has been achieved by their methods. Additionally, we also notice there are some valuable factors which can help us improve the performance of slot filling, such as inference, external knowledge base[38], text regularization and entity linking.

In Surdeanu and Ji [89], participants of the English Slot Filling Track are also building their models based on techniques mentioned. More importantly, they also share some observations with respect to methodologies in the competitions which are aligned with most of researches and us. Firstly, although a high number of methods look promising, we can observe that distant supervision is a winner compared to bootstrapping based on their performance[89]. Distant supervision dominates the best three systems in the competition. Moreover, many teams have combined rule-based methods with distant supervision which achieved better results. Secondly, the top three groups are using query expansion techniques, which rephrases the query in different ways, to improve performance. This means query expansion will help us locate relations accurately. Thirdly, within document co-reference resolution, it is important to detect name entities as explained by Ji et al. [39]. Although our topic is slot filling and not name entity recognition, correctness in name entity detection is the basis of slot filling. Thus, improvements in name entity detection will also help us realize slot filling. Additionally, active learning, in which the concept is that we can trust machine learning models to choose what data they want to learn from, also provides competitive results by filtering valuable information in training sets. Active learning allows the model to select the subset data proactively instead of passively from unlabelled dataset to annotate and is a proven effective way to reduce noise in distant supervision[7]. Last but not least, Natural Language Inference (NLI), the task of determining if one given statement semantically entails other meanings or statements in natural languages, plays an important role in slot filling. However, errors in NLI phrase bring more issues in slot filling based on experiments in Roth et al. [80].

2.1.4 Summary

The ways of human-thinking and human-understanding are based on their experiences and previous knowledge about the world. Sometimes, we need to evaluate all text content before we can fully understand the key information of an article. Pattern matching methods, such as bootstrapping, are attempting to use previous knowledge to find instances of relations in a sentence. Knowledge base methods, like distant supervision, focus on statistics and probability experiences for relation extraction in sentences level. Researchers pay efforts on leveraging knowledge bases, existing patterns and graph mining to detect complicated relations in texts. An overview of process, resources and outcomes can be found in Figure 2.1.

Based on the aforementioned observations, we are most interested in distant supervision with the help of other techniques such as Name Entity Recognition and Co-reference Resolution. The reason for this is threefold. First of all, distant supervision shows competitive results[87, 89]. A large amount of research has been done to improve it due to its popularity. Secondly, knowledge-based methods will be more applicable in different domains compared to pattern-based methods. This provides more potential for commercial applications. Thirdly, as our goal is to detect relations with the least labels, distant supervision, as an unsupervised method, meets our requirements. Compared to Open IE, it generates target relations directly. It saves our efforts on mapping annotations into our interesting relations. Considering all these facts, we will take distant supervision as the approach for unsupervised annotating relations. More variants of distant supervision will be elaborated in the experiment section.



Category	Approach	Strength	Limitations	Reference
Knowledge-based Approach	Distant Supervision (unsupervised)	• No manual work • No overfitting • No domain dependency • Promising results	• False positive due to the assumption	14 papers
Pattern-based Approaches	Bootstrapping (semi-supervised)	• No overfitting	• Need manual work. • Low precision, • semantic drift, • error propogation	10 papers
	Open Information Extraction (unsupervised)	• No relation restriction	• Duplicate information in annotation • Low recall	6 papers and testing on 3 systems
Other Approches	Combinations	• Better performances	• Troublesome implementation	9 papers

Figure 2.1: The Overview of Related Work of Annotation Approaches

2.2 Mainstream Approaches of Detecting Relations

For the purpose of clarity, we categorize previous approaches into the following groups:

- **Relation Extraction based on pre-identified entities**
  In this category, the process of the majority of existing methods can be split into two parts. The first part is to detect name entities of an annotated dataset. Then, based on the pre-identified entities, we detect relations from unstructured data which will help us to build static knowledge graphs. These methods can either be unsupervised or supervised.

- **Joint Extraction of entities and relations**
  The joint model first conducts entity recognition and then predicts relations between extracted entities, capturing the linguistic dependencies between entities and relation instances.

The main difference between these two groups is that non-joint models takes name entity recognition as a dependency while joint models detect both entities and relations at the same time. The latter one not only focuses on relation extraction, it considers connections and interactions between entities and relations.

2.2.1 Relation Extraction Based on Pre-identified Entities

Researchers made great progress in order to identify relations between a pair of pre-identified entities. The first step, Name Entity Recognition(NER), could be regarded as candidate generation before candidate validation[81]. Previous research can be categorized into two groups. In feature-based methods, researchers have been working on feeding features into machine learning classifiers for relation extraction. In the TAC KBP English Slot Filling track, a evaluation campaign for the extraction of 41 different relations, Lange Di Cesare et al. [44] investigated how statistical features, lexical features, name-entity features and syntactic features affect the performance of relation extraction. The authors in [81] applied distant supervision support vector machine for relation extraction. Most models in the TAC KBP English Slot Filling track were machine learning models which were highly dependent on feature engineering. According to their results[44], efficient feature extraction improved global precision and F1-score. However, this also proved that poor features and dependencies may lead to unsatisfied results of relation extraction.

In neural-based approaches, researchers worked on neural networks in order to address the current difficulty of extracting high-quality features in NLP. Zeng et al. [104] used convolutional deep neural networks to extract relations, which outperformed state-of-the-art methods in 2014. Zeng et al. [105] avoided feature engineering and instead developed a convolutional architecture with piece-wise max pooling to automatically extract useful features. The input of their model was the sentence and two pre-identified entity mentions. Similarly, developing the best-performing model in the SemEval 2010 relation classification task, Xu et al. [100] extracted the most relevant information from the shortest dependency path between two entities with the help of multichannel recurrent neural networks and long short term memory (LSTM) units.

2.2.2 Joint Extraction of entities and relations

Instead of regarding entity extraction as one of the assumptions, joint extraction methods consider the extraction of entities and relations at the same time. The integration of two sub-tasks is referred to as end-to-end relation extraction. It is not hard to understand that the performance of entity recognition has a great influence on relation extraction. More importantly, the performance of relation extraction could affect entity recognition. Early research such as the publications by Choi et al. [15] and Gupta et al. [31] mentioned that considering dependencies among entities and relations boost the performance of the joint model itself and also improve the performance of the independent tasks of entity recognition and relation extraction. Specifically, for the relation type of *employee of*, the first entity is supposed to be a person and the second entity should be an organization. However, addressed by Finkel et al. [26], these entity type preferences are commonly ignored by the popular distant supervision model developed by Mintz et al. [57]. Ignoring common sense rules leads to error in relation extraction. To address this issue, researchers have been putting more efforts into researching joint extraction of entities and relations.

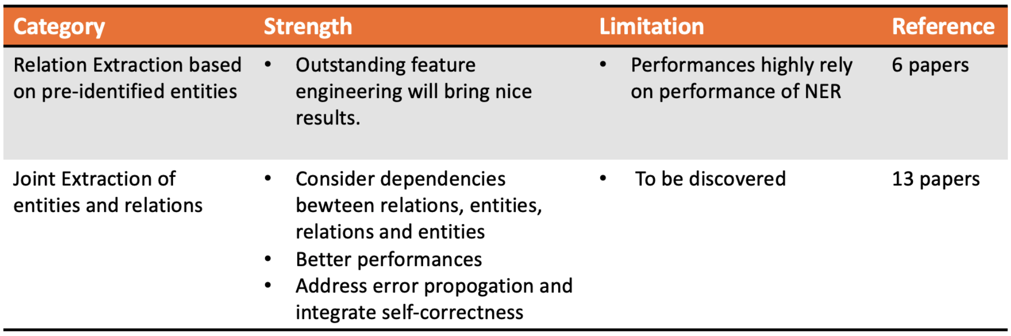
Finkel et al. [26] built a model which is able to capture interactions between relations and entities, resulting in 13% precision improvement over the baseline. Miwa and Bansal [58] combined valuable information like word sequence and dependency tree substructure by stacking different versions of bidirectional long short term memory (LSTM)-recurrent/recursive neural networks (RNNs), resulting in the best performance of nominal relation classification in the competition of SemEval-2010 Task 8. The model pre-trained the entity model and corrected labels at a later point to solve the problem of error propagation in early stages of training. Li and Ji [47] applied a structured perceptron[17] to model the interactions between entities and relations for relation extraction. Their end-to-end system achieved the best results in the Automatic Content Extraction (ACE) corpora.

Li et al. [45] followed the same path to realise relation extraction from biomedical texts. They applied Convolutional Neural Networks (CNN) as encoder to extract information from character level. They then stacked LSTM based on RNN (Bi-LSTM-RNN) to generate representations of target entities and their relation.

As parameters are shared between LSTM and RNN, parameters are tuned based on both performance of entity recognition and relation extraction. Other neural-based promising models can be found in Gupta et al. [31], Zheng et al. [110], Ren et al. [74], Zheng et al. [111] and Zeng et al. [106].

2.2.3 Summary

Traditional relation extraction models took extracted and/or handcrafted features as input to train various classifiers. Further neural-based methods built different neural networks to extract relations according to marked entity mentions by dependencies. Compared to separating relation extraction into multiple local classification problems, joint models consider long distance information, cross-task dependencies[47] and interactions between entity labels and relation labels. The latter mimics how people understand text according to their knowledge base and all information of texts from both character, sentence and document levels. Also, joint models of extracting entities and relations addressed the problem of error propagation. They improved the ability of self-correctness of models through validating entities and relations in an interactive way. An overview of the clear memorization and comparisons can be found in Figure 2.2.



Category	Strength	Limitation	Reference
Relation Extraction based on pre-identified entities	• Outstanding feature engineering will bring nice results.	• Performances highly rely on performance of NER	6 papers
Joint Extraction of entities and relations	• Consider dependencies bewteen relations, entities, relations and entities • Better performances • Address error propogation and integrate self-correctness	• To be discovered	13 papers

Figure 2.2: The Overview of Related Work of Relation Extraction Models

2.3 Gap Investigation

According to the aforementioned observations, we are able to draw the following conclusions. First of all, compared to Open IE, bootstrapping and other methods, distant supervision is a wise choice for automatic relation annotations as it is domain independent and shows promising results. With the help of techniques like co-reference resolution and NER, automatic annotation can be realized with satisfying performance. Secondly, as for relation extraction, the joint model has the ability of integrating valuable information from both entities and relations level which achieves better performance in general. Most importantly, it remediates the error propagation in the early phase of relation extraction. The joint model is more aligned with how people think and understand text. Thus, in this project we will start with distant supervision for dataset annotation and investigate joint models of entity and relation extraction. Details of this will be covered in next chapter.

However, we also notice something missing in the existing literature. On the one hand, in terms of complementary techniques like co-reference resolution, NER and query expansion, researchers have attempted to integrate some of them for the sake of improving the performance of relation extraction. But some questions have not been answered: having limited resources and time, which complementary methods should we focus on? Or which are the relative influential ones? How does the performance of one of the complementary methods affect the overall performance of relation extraction? What are the best combinations of these parameters? Investigating influential parameters of relation extraction will be an interesting direction of this project that could provide novel insights.

On the other hand, most researchers assumed that the relation between two entity mentions is binary. Intuitively, this is not the case in reality. For example, Amsterdam can be both ”birth place”, ”visited place” and/or ”working place” of another person at the same time. Simply assigning single relation to a pair of entities is bound to result errors in relation extraction. Consequently, in this thesis, we will be focusing on models of extracting complicated relation such as multiple relations between a pair of entities or various relations between several entities.

In summary, this chapter introduced a clear overview of annotation methods and models of relation extraction. Moreover, we have elaborated on how we investigate the research directions and have identified the gaps between current research and our expectations. Following the directions addressed, we will describe our experiments and answer research questions in the following chapter.

Chapter 3

Methodology

This chapter elaborates how to choose the dataset and benchmark model and construct the pipeline of this thesis. The structure of this chapter is briefly explained in Figure 3.1.

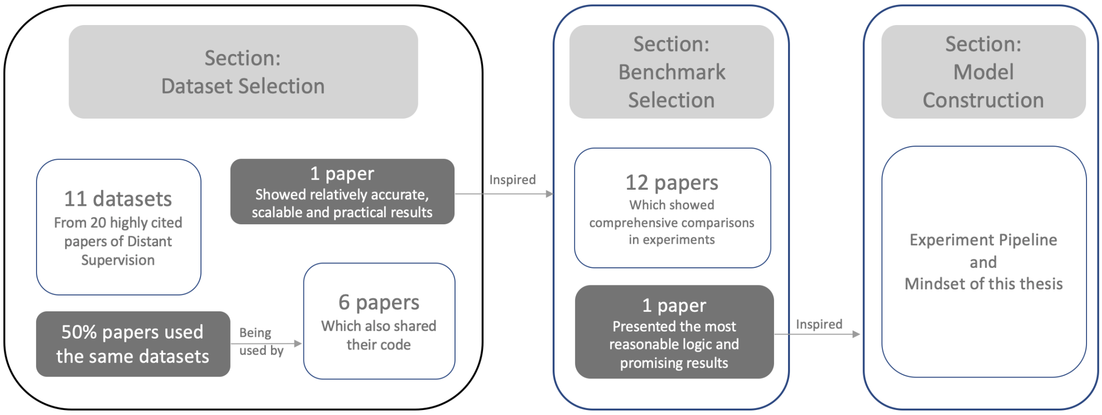

Section:
Dataset Selection
11 datasets
From 20 highly cited papers of Distant Supervision
1 paper
Showed relatively accurate, scalable and practical results
50% papers used the same datasets
Being used by
6 papers
Which also shared their code
Inspired
Section:
Benchmark Selection
12 papers
Which showed comprehensive comparisons in experiments
1 paper
Presented the most reasonable logic and promising results
Inspired
Section:
Model Construction
Experiment Pipeline
and
Mindset of this thesis

Figure 3.1: The Overview of the Methodology Chapter

3.1 Data Collection

3.1.1 Analysis of Candidate Datasets

This section includes the process of selecting candidate datasets. A number of reasons for the selection of these candidates will be gone over. In short, in order to filter the relevant datasets, we followed the following four steps:

- Step 1: Searched with query ’distant supervision’ in Google Scholar. As mentioned in the previous chapter, to automatically annotate datasets, distant supervision stands out from existing methods.

- Step 2: Picked the twenty most highly cited and most recent papers which have a focus on relation extraction.
- Step 3: Performed an analysis of candidate datasets.
- Step 4: Made a decision of what datasets to select for this project.

The first outcome of these four steps is a clear overview of candidate datasets in Table 3.1, 3.2, 3.3 and 3.4. In these tables, the full title, published year, authors, name, cost and resources of the dataset can be found. Based on the information gathered in these tables, we first provide the distribution of used datasets in Figure 3.2. Combining these with statistics in Table 3.1, 3.2, 3.3 and 3.4, we found that the dataset published by Riedel et al. [75] is a commonly used dataset in the field of relation extraction[36, 105, 74]. Among the twenty papers, twelve papers have used this dataset[75]. The total number does not equal twenty for the reason that some papers use more than one dataset.

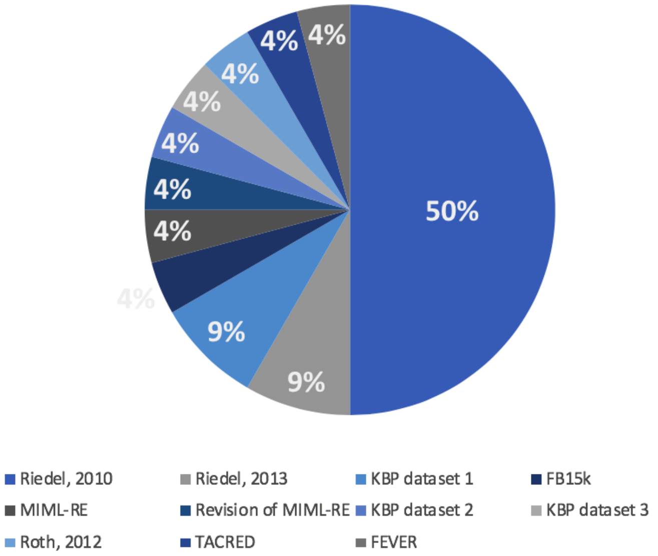


Figure 3.2: The Distribution of Used Datasets

After the collection of datasets, we went through each dataset and analyzed its characteristics. According to the usage frequency of the datasets, we can conclude that the most commonly used dataset was developed by Riedel et al. [75] through aligning Freebase[1] relations with the New York Times (NYT) corpus. The authors used the Stanford named entity recognizer[26] to find entity mentions in text and constructed relation mentions between entities within the same sentence. Based on this observation, we filtered out papers that are related to the dataset of Riedel et al. [75] and who provide publicly available code. The filtered results were published by: Fan et al. [24], Surdeanu et al. [92], Feng et al.

[1] https://developers.google.com/freebase/

[25], Hoffmann et al. [35], Ren et al. [74], Lin et al. [50]. After having conducted detailed analysis of these papers we have made some significant observations.

Observation 1

Firstly, the dataset by Riedel et al. [75] is in the format of *Protocol Buffers*[96], which is an extensible and efficient mechanism for serializing structured data developed by Google. Hoffmann et al. [35] and Surdeanu et al. [92] offer detailed descriptions of their own model and the model from Riedel et al. [75]. Consequently, most researchers are using the dataset by Riedel et al. [75], processed by Hoffmann et al. [35] and Surdeanu et al. [92]. This is the reason for why most researchers solely mention the usage of the data by Riedel et al. [75], and omit an explanation of the pre-processing steps.

Observation 2

Secondly, Feng et al. [25] has conducted experiments based on the implementation by Lin et al. [50]. Lin et al. [50] have built a Convolutional Neural Network (CNN) with sentence-level attention to reduce noise in texts. Their model achieves better performance than other neural network approaches and feature-based methods. Leveraging the code released by Lin et al. [50], Feng et al. [25] re-implemented the sentence-level classification model[104], the bag-level model[105] and the model from Lin et al. [50] as baselines. Then Feng et al. [25] developed their sentence-level model using a reinforcement learning framework that outperforms the state-of-the-art baselines[50] in relation classification. Accordingly, it is reasonable to exclude the publication by Lin et al. [50] from the list. The model of Feng et al. [25] is not ideal when the data is noisy, but this is common in practice.

Observation 3

Thirdly, Ren et al. [74] and Fan et al. [24] both built models focusing on reducing the impact of noise in relation extraction and obtained competitive results. However, one of the limitations of Fan et al. [24] is that their model is unable to process new testing data. This is a result of the fact that in this case it is needed to reconstruct the feature matrix which is computed in an iterative fashion. Consequently, this limits the amount of applications their model has in real life. The distant supervision model by Ren et al. [74] is noise-robust and more domain-independent. They make use of the joint extraction of entities and relations in text combined with distant supervision. This means that their model can extract not only name entities with high accuracy, but also relations with impressive F1-scores. Additionally, they re-implement eight state-of-the-art relation extraction models together with *LSTM*[28] and *Bi-GRU* architectures for the sake of comparison. Experimental results show their model achieves its robust performance across corpora of various sizes and has high scalability which enables processing the full-size dataset.

3.1.2 Summary

In summary, the papers of Hoffmann et al. [35] and Surdeanu et al. [92] which were published several years ago, act as a foundation in this field, which provides the clean and processed datasets with other researchers. Based on previous contributions, Feng et al. [25] proposed an advanced version based on the implementations from Lin et al. [50]. Furthermore, Ren et al. [74] shows high scalability and practicality compared to Fan et al. [24]. Ren et al. [74] applied distant supervision as an annotation method to create an automatically annotated dataset based on datasets from Riedel et al. [75]. Considering the characteristics of being generally applicable, domain-independent, noise-robust and effective together with our observations in previous chapter, we have decided to take the paper by Ren et al. [74] as an entry point of distant supervision for relation extraction using the dataset published by Riedel et al. [75] for this project.

[2] https://github.com/davidsbatista/Annotated-Semantic-Relationships-Datasets/blob/master/datasets/kbp37-master.zip

[3] http://www.kozareva.com/downloads.html

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

[5] http://nlp. stanford.edu/software/mimlre.shtml

[6] https://github.com/thunlp/KB2E

Idx	Title	Year	Authors	Dataset	Dataset related paper	Code	Cost	Download
1	Effective slot filling based on shallow distant supervision methods[80]	2014	Saarland University	Roth, 2012.	Generalizing from Freebase and Patterns using Cluster-Based Distant Supervision for KBP Slot-Filling[79]	No	-	Not found
2	Relation classification via recurrent neural network[108]	2015	Tsinghua University	Revision of MIML-RE	Combining distant and partial supervision for relation extraction[7]	Yes	Free	Github[2]
				SemEval-2010 Task 8	Multi-way classification of semantic relations between pairs of nominals[33]	Yes	Free	Github[3]
3	Distant Supervision for Relation Extraction with Sentence-Level Attention and Entity Descriptions[36]	2017	National Laboratory of Pattern Recognition in China	**Riedel, 2010**	Modeling relations and their mentions without labeled text[75]	No	Free	Link[4]
4	Combining distant and partial supervision for relation extraction[7]	2014	Stanford University	MIML-RE	Combining distant and partial supervision for relation extraction[7]	Yes	Free	Github[2], Link[5]
5	Distant supervision for relation extraction via piecewise convolutional neural networks[105]	2015	National Laboratory of Pattern Recognition in China	**Riedel, 2010**	Modeling relations and their mentions without labeled text[75]	No	Free	Link[4]
6	Label-Free Distant Supervision for Relation Extraction via Knowledge Graph Embedding[98]	2018	College of Computer Science and Technology, Zhejiang University,	**Riedel, 2010**	Modeling relations and their mentions without labeled text[75]	No	Free	None
				FB15k	Learning Entity and Relation Embeddings for Knowledge Graph Completion[49]		Free	Github[6]

Table 3.1: Candidate Dataset

Idx	Title	Year	Authors	Dataset	Dataset related paper	Code	Cost	Download
7	Distant supervision for relation extraction with an incomplete knowledge base[56]	2013	New York University & IBM	**Riedel, 2010**	Modeling relations and their mentions without labeled text[75]	Partly	Free	Link[4]
				KBP dataset1	Multi-instance multi-label learning for relation extraction[92]		Free	Github[5]
8	Distant supervision for relation extraction with matrix completion[24]	2014	Tsinghua University	**Riedel, 2010**	Modeling relations and their mentions without labeled text[75]	Yes	Free	Link[4]
				Riedel, 2013	Relation extraction with matrix factorization and universal schemas[76]		-	Not found
9	Infusion of labeled data into distant supervision for relation extraction[66]	2014	New York University	KBP dataset2	Stanford's Distantly-Supervised Slot-Filling System.[91]	No	-	Not found
					Knowledge base population: Successful approaches and challenges[37]		-	Not found
10	Multi-instance multi-label learning for relation extraction[92]	2012	Stanford University	**Riedel, 2010**	Modeling relations and their mentions without labeled text[75]	Yes	Free	Link[4]
				KBP dataset1	Multi-instance multi-label learning for relation extraction[92]		Free	Github[5]
11	Relation extraction with matrix factorization and universal schemas[76]	2013	University College London; University of Massachusetts at Amherst	Riedel, 2013	Relation extraction with matrix factorization and universal schemas[76]	No	-	Not found

Table 3.2: Candidate Dataset

[7] https://github.com/JuneFeng/RelationClassification-RL

[8] https://catalog.ldc.upenn.edu/LDC2018T24

[9] http://fever.ai/resources.html

[10] http://raphaelhoffmann.com/mr/

[11] https://github.com/INK-USC/DS-RelationExtraction

[12] https://github.com/xiaoling/figer

[13] http://mars.cs.utu.fi/BioInfer/

Idx	Title	Year	Authors	Dataset	Dataset related paper	Code	Cost	Download
12	Modeling relations and their mentions without labeled text[75]	2010	University of Massachusetts	**Riedel, 2010**	Modeling relations and their mentions without labeled text[75]	No	Free	Link[4]
13	Reinforcement Learning for Relation Classification from Noisy Data[25]	2018	Microsoft Research Asia; Tsinghua University	**Riedel, 2010**	Modeling relations and their mentions without labeled text[75]	Yes	Free	Link[4], Github[7]
14	Position-aware attention and supervised data improve slot filling[109]	2017	Stanford University	TACRED	Position-aware attention and supervised data improve slot filling[109]	Yes	$25	Github[8]
15	FEVER: a large-scale dataset for Fact Extraction and VERification[94]	2018	Amazon Research Cambridge University of Sheffield	FEVER	FEVER: a large-scale dataset for Fact Extraction and VERification[94]	Yes	Free	Link[9]
16	Knowledge-based weak supervision for information extraction of overlapping relations[35]	2011	University of Washington	**Riedel, 2010**	Modeling relations and their mentions without labeled text[75]	Yes	Free	Link[10]
17	CoType: Joint Extraction of Typed Entities and Relations with Knowledge Bases[74]	2017	University of Illinois at Urbana-Champaign	**Riedel, 2010**	Modeling relations and their mentions without labeled text[75]	Yes	Free	Github[11]
				KBP dataset3	Fine-Grained Entity Recognition[51]	Yes	Free	Github[12]
				BioInfer	BioInfer: a corpus for information extraction in the biomedical domain[71]	Yes	Free	Link[13]

Table 3.3: Candidate Dataset

Idx	Title	Year	Authors	Dataset	Dataset related paper	Code	Cost	Download
18	SEE: Syntax-aware Entity Embedding for Neural Relation Extraction[32]	2018	Alibaba Group	**Riedel, 2010**	Modeling relations and their mentions without labeled text[75]	No	Free	Github[14]
19	Learning Entity and Relation Embeddings for Knowledge Graph Completion[49]	2015	Tsinghua University & Samsung R&D Institute	FB15k and WN18	Learning Entity and Relation Embeddings for Knowledge Graph Completion.[49]	Yes	Free	Github[6]
20	Neural relation extraction with selective attention over instances[50]	2016	Tsinghua University	**Riedel, 2010**	Modeling relations and their mentions without labeled text[75]	Yes	Free	Github[15]

Table 3.4: Candidate Dataset

The main points addressed in this section are the selection of our benchmark model and how this model works. Moreover, considering the paper published by Ren et al. [74] as our entry point, we will examine the key ideas of this paper and other papers which have referred to it.

3.2 Benchmark Selection

The original training corpus contains 1.18 million sentences extracted from around 294k New York Times news articles from 1987 to 2007 [75]. Ren et al. [74] first built a distant supervision model for mapping detected entities to entities in a knowledge base. Here they applied *Stanford CoreNLP Parser*[16] to generate name entity tags. Based on name entity tags, they filtered candidate entity mentions for distant supervision. As previously mentioned, the main idea of distant supervision is mapping entities and relations into entities and relations in a knowledge base. For every pair of entity mentions in the same sentence, the authors labeled it with corresponding entities and relations if they found any in the knowledge base. For those entities pairs which could be mapped back to the knowledge base, we will refer to as linkable entities. Entity pairs that cannot be mapped will be referred to as unlinkable entities. The last step of the distant supervision segment, following Hoffmann et al. [35], was to extract 30% of the unlinkable entities as negative samples, labelled as 'None' in the training dataset. Additionally, they took 395 sentences which were manually annotated by Hoffmann et al. [35] as test data.

[14] https://github.com/SUDA-HLT

[15] https://github.com/thunlp/NRE

[16] https://stanfordnlp.github.io/CoreNLP/corenlp-server.html

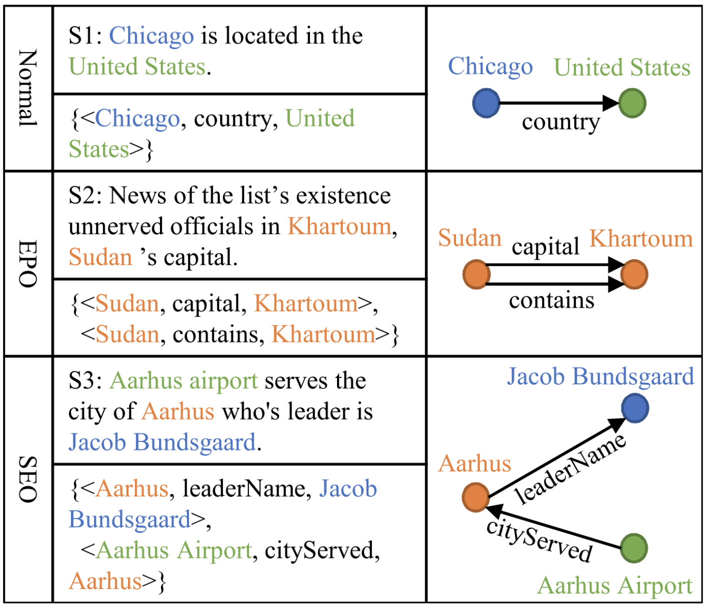
After the annotation of the dataset through distant supervision, the authors worked on the joint model of extracting entities and relations. Instead of using common word embedding, the joint model, called CoType, focused on constructing their own embeddings, including entity/relation mention embeddings, feature embeddings and relation/entity types embeddings. Combined with their own loss function, CoType could model types of entity and relation and interactions between entity and relation. The CoType framework achieved the best results of relation extraction when it was published.

The research directions that Ren et al. [74] investigated are in line with the observations in our literature review. Combining distant supervision with the joint model of entity and relation extraction it is possible to build a pipeline for relation extraction without manual annotations but with competitive performance. Upon further investigation of the research direction of Ren et al. [74], we found several interesting papers in the papers which were inspired by the CoType framework.

Regarding Ren et al. [74] as one of their baseline models, Zheng et al. [112] created a novel tagging schema to jointly extract entities and relations, which achieved better results on the same dataset used by Ren et al. [74]. Moreover, Zheng et al. [112] addressed the limitation of CoType framework. As is the case with other publications in the field of relation extraction [48, 59], Ren et al. [74] required complicated feature engineering (self-defined embeddings). Consequently, following Miwa and Bansal [58] and Vaswani et al. [97], Zheng et al. [112] also applied bi-directional Long Short Term Memory (Bi-LSTM) for encoding which reduced the size of the heavy feature engineering task. In order to prove the effectiveness of their methods, the authors re-implemented several pipeline models mentioned in our related work such as CoType [74], MultiR [35] and DS-logistic [57]. Based on the experiment results it could be observed that end-to-end models with Bi-LSTM encoding outperformed traditional approaches.

Similarly, taking CoType (Ren et al. [74]), MultiR (Hoffmann et al. [35]), DS-logistic (Mintz et al. [57]), Line (Tang et al. [93]) and DS-Joint (Li and Ji [48]) as benchmark models, Wang et al. [99] considered not only interactions between entities and relations but also dependencies between relations by their graph schema. They also used Bi-LSTM for the acquisition of valuable information in text. More importantly, they noticed the same problem we addressed in our literature review. Previous state-of-the-art systems obtained impressive results for the reason that their models considered dependencies between entities and relations. However, ignoring associations between relations will miss overlapping relations in sentences. For example, for the sentence of 'Vincent Willem van Gogh was born in Groot-Zundert, Netherlands', Zheng et al. [112] may only be able to extract the relation of 'birth place' between 'Vincent Willem van Gogh' and 'Groot-Zundert' instead of the relation of 'contain' between 'Netherlands' and 'Groot-Zundert'.

The overlapping relation issues were also elaborately addressed by Zeng et al. [107]. Based on our literature review, we can gather that it is commonly ignored by the majority of previously conducted research. In the paper of Zeng et al. [107], the authors categorized annotated relations into *Normal Relations*, *EntityPairOverlap Relations* and *SingleEntityOverlap Relations*. A relation belongs to the *Normal Relations* category when there exists only one relation between an entity pair. A relation is part of the *EntityPairOverlap Relations* category when two relations share the same entity pair. Relations containing two



Normal	S1: Chicago is located in the United States.	
	{<Chicago, country, United States>}	
EPO	S2: News of the list's existence unnerved officials in Khartoum, Sudan 's capital.	
	{<Sudan, capital, Khartoum>, <Sudan, contains, Khartoum>}	
SEO	S3: Aarhus airport serves the city of Aarhus who's leader is Jacob Bundsgaard.	
	{<Aarhus, leaderName, Jacob Bundsgaard>, <Aarhus Airport, cityServed, Aarhus>}	

Figure 3.3: An Example of Overlapping Relations[107]

relations which share only one entity belong to the *SingleEntityOverlap Relation* category. For instance, in the sentence of 'In 1916, Maugham moved to South Pacific to initial his novel The Moon and Sixpence', 'Maugham' is the writer of the book ' The Moon and Sixpence'. Also 'Maugham' has 'visited place' relation of the entity mention of 'South Pacific'. In this case, these two relations share the same entity of 'Maugham' so these three entities and corresponding relations should belong to *SingleEntityOverlap Relation*. Examples can also be found in Figure 3.3.

Previous work only allowed one label for one entity, which resulted in a large amount of relations that were contained in the text being missed. To address this problem, Zeng et al. [107] applied a copy mechanism which enables entities to be copied several times according to the number of relations related to this entity. This observation is in line with our own, which states that complicated relations exist between entities. While other researchers built excellent models, their assumption of the existence of at most one relation for two entities is not reasonable. In fact, this might be one of the reasons why relation extraction is still far away from being solved, which we also mentioned in the gap investigation of the previous chapter. This is the first point that has driven us to choose the paper by Zeng et al. [107] as a baseline model for this thesis.

Secondly, again being aligned with our observations, Bi-LSTM was used in Zeng et al. [107] as an encoder for extract information from sentences. Compared to model infrastructures, it showed its ability of capturing valuable information of content. Even considering tagging and graph schemas also have shown competitive results, it is unnecessary to put efforts towards additional labelling or tagging if we are able to avoid complicated feature engineering completely.

Last but not least, it is a challenge to identify the state-of-the-art approach in the area of relation extraction. When researchers evaluate their approach by standards that deviate from others, such as in the use of datasets and/or target relations, their models are not comparable. Moreover, some models have been shown to outperform other baseline models, but from one specific perspective. It is not realistic to re-implement all existing models for scientific comparisons. However, following CoType [74], Zheng et al. [112] compared their own models with a multitude of baseline models (MultiR [35], DS-logistic [57], Line [93] and DS-Joint [48]). These systematic comparisons made their results more reliable and convincing. Each of the listed models used the same dataset for training: the New York Times news corpus automatically annotated by Ren et al. [74]. Zeng et al. [107] regarded the previous state-of-the-art approach by Zheng et al. [112] as the baseline model.

Taking the aforementioned points into account, we will use the model developed by Zeng et al. [107] as the baseline model in this thesis. The reason for this is threefold. First, the paper addressed the fact that more than one relation should be detected for a pair of entities. Secondly, it applied Bi-LSTM as an encoder in order to generate vectors as the representation of the semantic meaning of a sentence. This approach has been proven to be one of most effective models for relation extraction. Lastly, Zeng et al. [107] built a reliable state-of-the-art system as they compared their model to a series of previously introduced approaches. We will further elaborate on this baseline model in the next section.

3.2.1 Benchmark Introduction

A bird view of the methodology of Zeng et al. [107] can be seen in Figure 3.4.

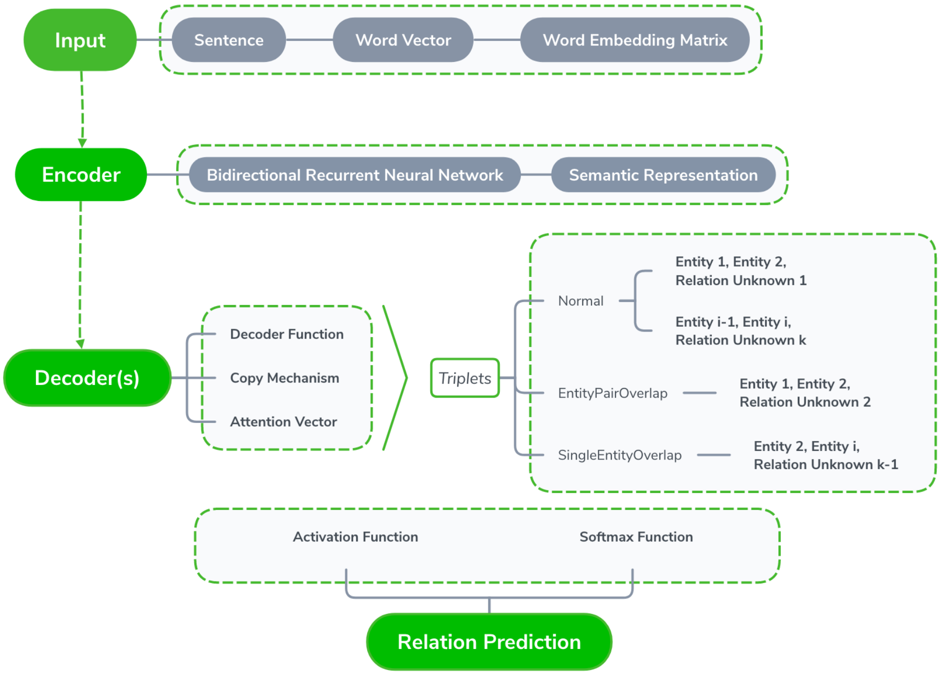


Figure 3.4: The Structure of Benchmark Model

Taking sentences as input, the first step of the process is to extract the word embeddings by providing one of the sentences as input for the encoder. The output of the encoder is a fixed-length vector representation which expresses the semantic meaning of the sentence. Next, the vector representation is fed into the decoder for generation of triplets for different relations using the copy mechanism and relation prediction. In this thesis, we will take the optimal parameters tuned by Zeng et al. [107] for the experiments. The clear overview of how encoders and decoders work has been shown in Figure 3.5. As we mentioned, A bi-directional RNN is responsible for encoding the input data and a decoder is targeting on generating triples, which include relation types and relation instances. It is based on the prediction results and the text copied from the input data. With this model, complex relation instances, such as more than one relation type between the same pair of relation entities, can be successfully extracted.

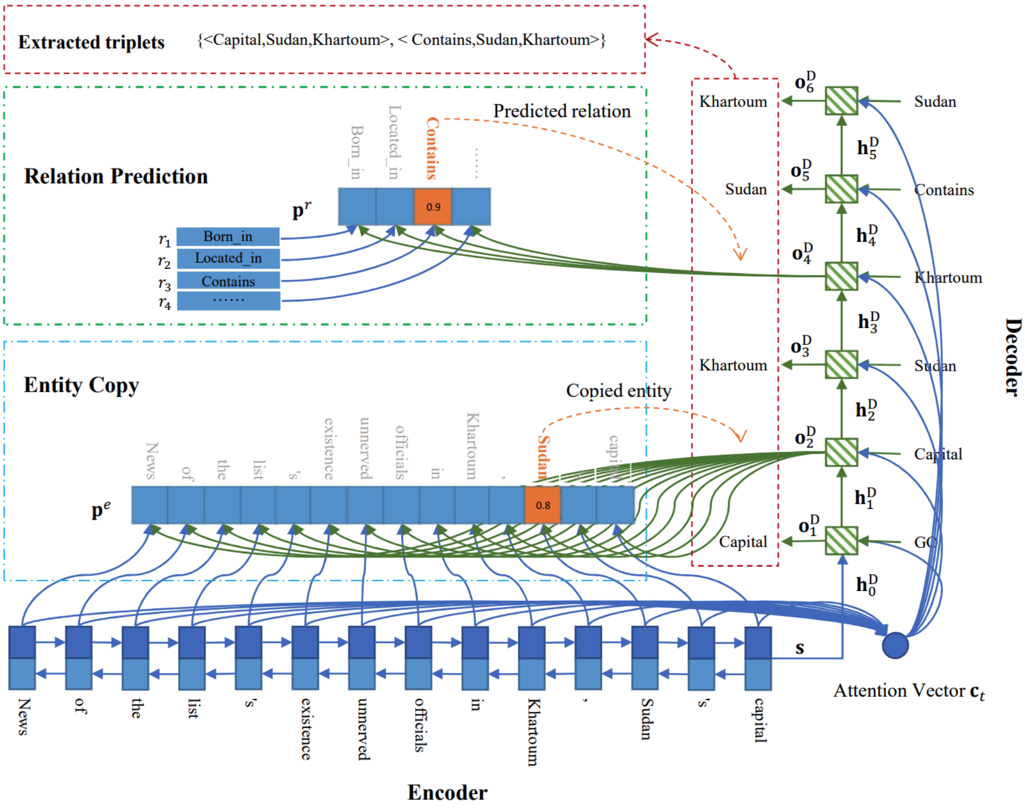


Figure 3.5: The Structure of the Encoder and the Decoder[107]

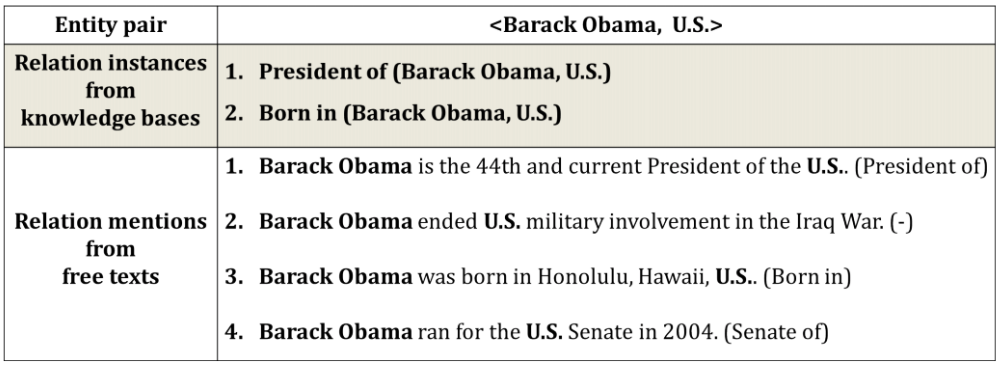
3.3 Model Construction

After reading this section, readers will be clear about how we build the model and what are supposed to be proven. The experiments will be conducted in the model to be explained. We will first introduce the pipeline of the model. Then we elaborate how we conduct experiments in the model.

3.3.1 Pipeline

Figure 3.7 contains an overview of our pipeline structure for relation extraction. The entire pipeline can be divided into two parts. In the first part, we followed the approaches by Ren et al. [74] and Wang et al. [99] to annotate the corpus using distant supervision. To provide an example, we included an annotated sentence in Figure 3.7 in which it is clearly explained how each sentence segment was annotated by the model. In the second part, we took the model designed by Zeng et al. [107] as a benchmark model for joint extraction of entities and relations.

The input of the pipeline is raw texts without annotations, which includes several paragraphs with interesting relations mentions and corresponding entities mentions. Following steps mentioned in the Distant Supervision block of Figure 3.7, we can locate detected entity and relation mentions in KB.



Entity pair	**<Barack Obama, U.S.>**
Relation instances from knowledge bases	**1. President of (Barack Obama, U.S.)** **2. Born in (Barack Obama, U.S.)**
Relation mentions from free texts	1. **Barack Obama** is the 44th and current President of the **U.S.**. (President of) 2. **Barack Obama** ended **U.S.** military involvement in the Iraq War. (-) 3. **Barack Obama** was born in Honolulu, Hawaii, **U.S.**. (Born in) 4. **Barack Obama** ran for the **U.S.** Senate in 2004. (Senate of)

Figure 3.6: An Example of Data in Freebase[24]

As Figure 3.6 shows, for the entity pair [Barack Obama, U.S.], we can find several sentences describing him in the Wikipedia. Based on those descriptions, Freebase provides triples with different relations related to Barack Obama. So if we detect the entity pair [Barack Obama, U.S.] in one of sentences, we are able to annotate entity relation mentions according to similarity between the sentence to be annotated and relevant sentences in Freebase.

By doing this, we can easily annotate datasets according to ground truth provided by KB. In the center of Figure 3.7, we show an example of annotated sentence, where entity and relation mentions are detected and labelled. With this automatic annotated dataset, we

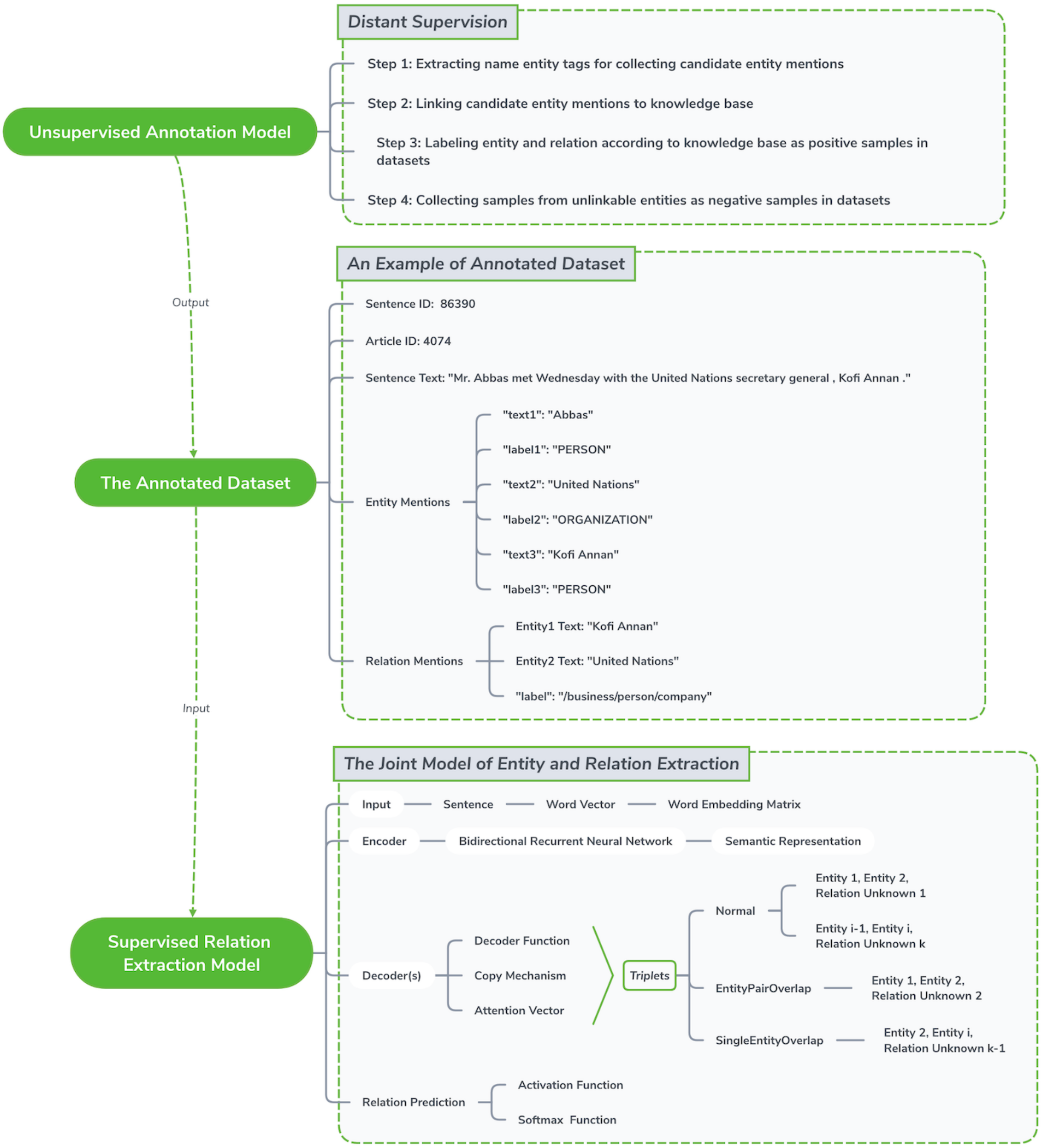

Distant Supervision
Step 1: Extracting name entity tags for collecting candidate entity mentions
Step 2: Linking candidate entity mentions to knowledge base
Step 3: Labeling entity and relation according to knowledge base as positive samples in datasets
Step 4: Collecting samples from unlinkable entities as negative samples in datasets
Unsupervised Annotation Model
Output
An Example of Annotated Dataset
Sentence ID: 86390
Article ID: 4074
Sentence Text: "Mr. Abbas met Wednesday with the United Nations secretary general , Kofi Annan ."
Entity Mentions
"text1": "Abbas"
"label1": "PERSON"
"text2": "United Nations"
"label2": "ORGANIZATION"
"text3": "Kofi Annan"
"label3": "PERSON"
The Annotated Dataset
Relation Mentions
Entity1 Text: "Kofi Annan"
Entity2 Text: "United Nations"
"label": "/business/person/company"
Input
The Joint Model of Entity and Relation Extraction
Input
Sentence
Word Vector
Word Embedding Matrix
Encoder
Bidirectional Recurrent Neural Network
Semantic Representation
Supervised Relation Extraction Model
Decoder(s)
Decoder Function
Copy Mechanism
Attention Vector
Triplets
Normal
Entity 1, Entity 2, Relation Unknown 1
Entity i-1, Entity i, Relation Unknown k
EntityPairOverlap
Entity 1, Entity 2, Relation Unknown 2
SingleEntityOverlap
Entity 2, Entity i, Relation Unknown k-1
Relation Prediction
Activation Function
Softmax Function

Figure 3.7: Experiment Pipeline

could move further for realizing supervised relation extraction. And the supervised relation extraction model has been explained in previous section.

3.3.2 Experiment Aspect

Based on Figure 3.7, some noteworthy observations can be made. These matters have previously been addressed by existing researches. However, none of the previously conducted studies compared their influences on relation extraction in the same pipeline. For this reason, in this thesis, we will fill in the existing gap of the evaluation of influential complementary approaches in relation extraction through a number of experiments. In this project, we have focused on four aspects:

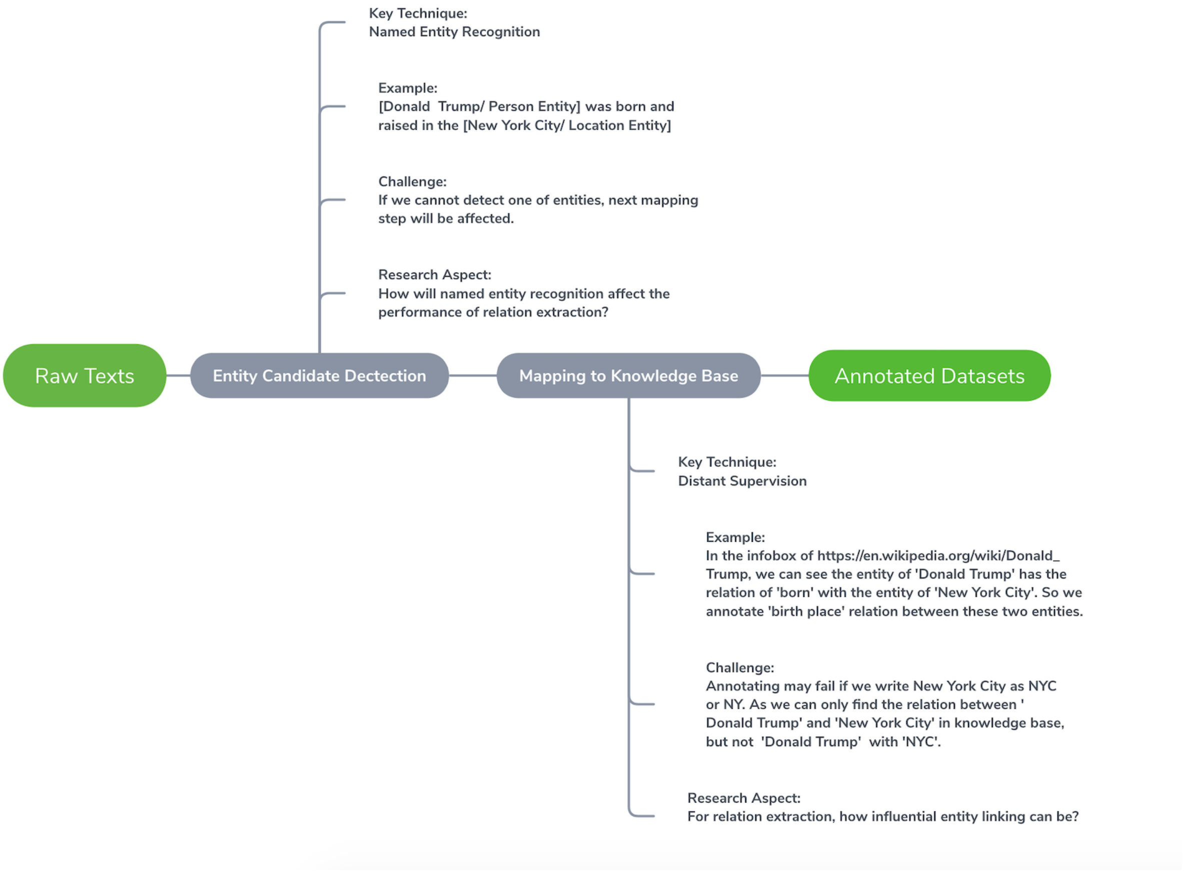

Key Technique:
Named Entity Recognition
Example:
[Donald Trump/ Person Entity] was born and raised in the [New York City/ Location Entity]
Challenge:
If we cannot detect one of entities, next mapping step will be affected.
Research Aspect:
How will named entity recognition affect the performance of relation extraction?
Raw Texts
Entity Candidate Dectection
Mapping to Knowledge Base
Annotated Datasets
Key Technique:
Distant Supervision
Example:
In the infobox of https://en.wikipedia.org/wiki/Donald_Trump, we can see the entity of 'Donald Trump' has the relation of 'born' with the entity of 'New York City'. So we annotate 'birth place' relation between these two entities.
Challenge:
Annotating may fail if we write New York City as NYC or NY. As we can only find the relation between ' Donald Trump' and 'New York City' in knowledge base, but not 'Donald Trump' with 'NYC'.
Research Aspect:
For relation extraction, how influential entity linking can be?

Figure 3.8: Experiment Design Part 1

Aspect 1: Named Entity Recognition

The annotation process starts by detecting entity candidates for annotation. As can been seen in Figure 3.8, the goal of detecting entity candidates is to find named entities such as *person*, *location* and *organization*. In the shown example sentence, ”Donald Trump” should be detected as *person* entity and ”New York City” as a *location* entity. With this process we intend to gain information about which words may be related to our defined relations. If we fail to successfully detect entities from the input text, relation prediction will be an impossible task. As a matter of fact, the quality of annotation is highly dependent on the performance of Named Entity Recognition as only detected entities will be taken into consideration when linking them with the knowledge base. When named entity recognition is done poorly, valuable training samples are lost.

Based on this observation, we have decided to study how named entity recognition affects the performance of relation extraction and how important it is compared to other components. In the baseline model [74], they used *Stanford CoreNLP*[54] for Named Entity Recognition. Recently, compared to excellent work by Peters et al. [67], Chiu and Nichols [14], Devlin et al. [20], Clark et al. [16] and Aguilar et al. [1] in the task of Named Entity Recognition, Akbik et al. [6] built the model, called *Flair*, is current state-of-the-art based on their experimental results. As the input of the pre-trained bidirectional character language model (marked as yellow in Figure 3.9), a sentence, also a character sequence, will be split into each word or token. And for each token, it can be converted into a contextual embedding by information carried in cell states of the first and last character. This word embedding is fed into a vanilla Bi-LSTM-CRF sequence labeling model (marked as blue in Figure 3.9), then named entities can be predicted.

Another popular and also the most well known Named Entity Recognizer is NLTK NER, which was first time introduced by Loper and Bird [52]. It stands for Natural Language Toolkit and is an open source program. As we mentioned, the baseline model used *Stanford CoreNLP* for NER in the process of automatic annotation. In our experiments, we will apply both *NLTK NER* and *Flair NER*. By comparing the results of relation extraction with different methods of NER, we will analyze how NER affects relation extraction and also how NER interacts with other component of relation extraction models.

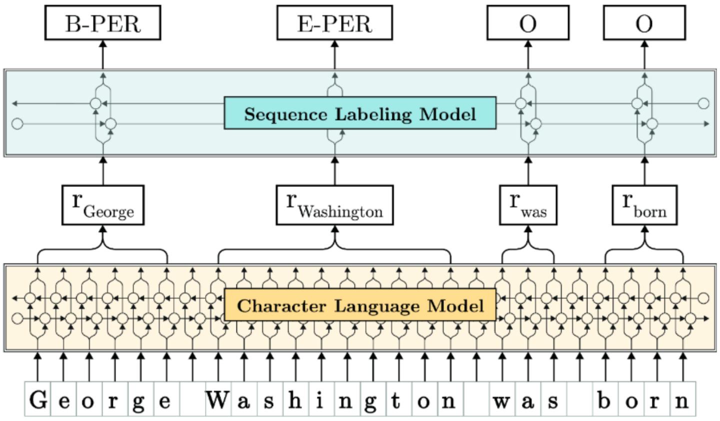


Figure 3.9: The Framework of Flair NER[6]

Aspect 2: Entity Linking

The second aspect is based on the observation that it is common to refer to a single entity with different expressions. Entity linking, commonly referred as named entity disambiguation or normalization[46], is the task which aims to link several entities to the one ground truth entity with reference to a knowledge base. For example, linking Sir Winston Leonard Spencer-Churchill and Prime Minister Churchill with Winston Churchill can be one example of entity linking. The Figure 3.10 also clearly explains how difficult entity linking and how confusing the model can be. Entity linking helps facilitate important fields of NLP such as KBP, information integration and question answering[86]. By successfully linking entities based on knowledge base, we can enrich semantics of entities[69] and help machine learning models understand entities.

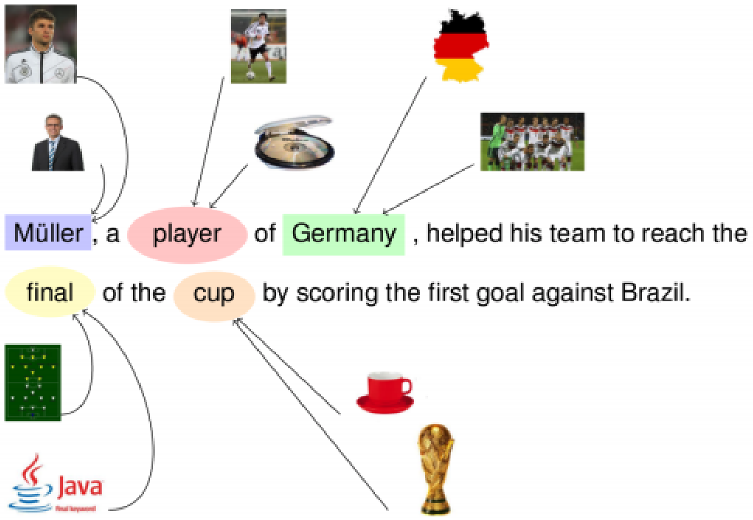

Müller, a player of Germany, helped his team to reach the
final of the cup by scoring the first goal against Brazil.
Java

Figure 3.10: The Visualization of the Difficulty of Entity Linking Tasks[27]

However, entity ambiguity and name variations are on of the main challenges in entity linking [86]. In text, full name, partial name, abbreviations and alternate spelling can be representative of the same named entity[86, 61]. For instance, we can call Barack Obama by his name or refer to him as ”the 44th President of the United States” or ”President Obama”. This is an example of name variations. In what way these methods affect relation extraction is not known and is therefore worth exploring.

The human mind is able to interpret that these three expressions are linked to the same entity. However, machine learning models may misinterpret this information. To aim to solve this problem, researchers made significant progress on entity linking, entity matching, named entity disambiguation and named entity normalization[60, 101, 69, 62, 86, 13, 22, 61, 27, 77, 113, 63, 95, 78]. In this project, we only consider the papers ([22, 61, 27, 77, 113, 63, 78, 95]) with available code for evaluating the impacts of entity linking in relation extraction. After carefully reading these papers and testing their code, we found some issues. First of all, for example Eshel et al. [22], setting up data and running experiments takes very long time and the needed disk space is at least 300GB. Secondly, papers such as Moussallem et al. [61] achieved impressive results in multilingual datasets but not English. As for the recent paper from Rosales-Méndez et al. [78], the provided interface obviously missed lots of annotations during several rounds of test. Therefore, we excluded papers, which consumed unreasonable long time during testing, failed to link relevant entities or do not applicable to English.

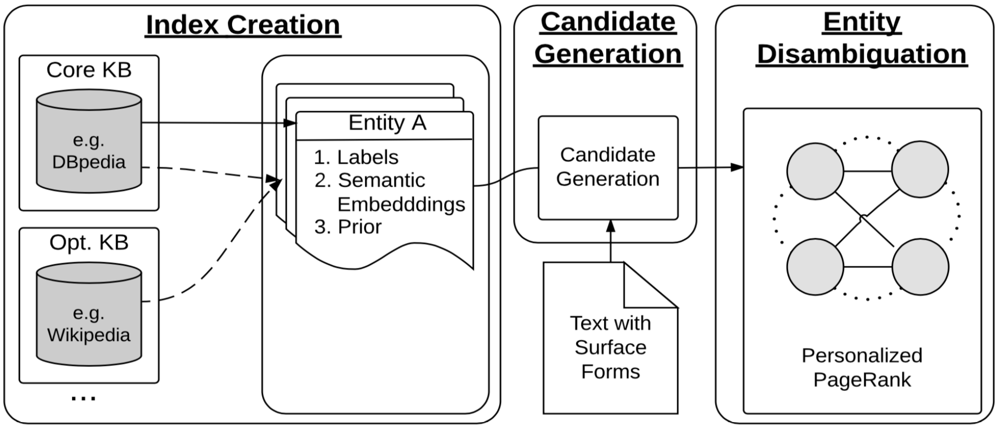


Figure 3.11: The pipelien of DoSer for Entity Disambiguation Tasks[113]

Ganea et al. [27] created a principled probabilistic graphical model which could explain the statistics of training set without complicated feature engineering. Usbeck et al. [95], Zwicklbauer et al. [113], Ngomo et al. [63] all used GERBIL[77], a general entity annotation system and annotation benchmark framework, to compare their results with existing models. The model AGDISTIS, built by Usbeck et al. [95] in 2014, was the state-of-the-art approach in terms of Named Entity Disambiguation. In 2016, DoSer, developed by Zwicklbauer et al. [113], outperformed AGDISTIS on seven public datasets using its generated semantic entity embeddings. In 2018, Ngomo et al. [63] introduced their automatic benchmark generator BENGAL. BENGAL applied RDF (Resource Description Framework) to generate annotations and analysed results based on the differences of POS annotations, F1-score of annotators and some other distribution features between manual annotated datasets and datasets annotated by BENGAL. However, we found that for some datasets, annotations by BENGAL has a high correlation with annotation in manual annotated datasets. Others are not highly correlated. This means the performances of BENGAL are unstable cross datasets. In this case, it will not be ideal to extra work on evaluating the similarity between our dataset and those datasets where BENGAL worked well.

We also tried DoSer as the model for entity disambiguation considering its simplicity and general disambiguation accuracy. DoSer divides the task of entity disambiguation into three steps: index creation, candidate generation and annotation. Index creation aims to define which entities need to be disambiguated. Also, it generates an embedding and a prior probability for each entity. In the step of candidate generation, the model define possible candidates for an entity form so that an entity candidate graph can be built. Moreover, making use of a PageRank algorithm, the model selects the disambiguated target entity with highest ranking score for a surface form. The details of DoSer can be found in Figure 3.11. However, after weeks efforts, we failed to run the source code of DoSer due to limited ram memory to set up Word2Vec Server.

To summarize, we filtered 14 relevant papers for the task of Entity Linking and neglected Moussallem et al. [61] and Ngomo et al. [63] due to its limited application areas. Moreover, we tested the models with public available code, such as Ngomo et al. [63], Rosales-Méndez et al. [78], Eshel et al. [22] and Zwicklbauer et al. [113]. Unfortunately, none of them was able to used in this thesis considering the poor performances during testing and our limited computational resources. Admittedly, the heaviness of the models during testing also showed the challenges of entity linking task. Still, there is a long way to go for machine to be as intelligent as human in nature language space. We will leave this topic for our future work.

Aspect 3: Length of Sentences in Training Set

The third aspect focuses on the length of sentences in training set. In this part of experiments, we planed to gradually exclude documents with very long sentences from training set and test in the same test test. A clear visualisation has been made, which can be found in Figure 3.12. It is known by common sense that human can remember the texts and understand the contexts better, when the document is shorter and has straightforward logic. For neural networks, the usage of memory gates play the important role in considering contextual background and carrying the information while making decisions. Shorter sentences in documents normally are with simple logic and tend to have fewer complicated relation instances.

Usually, researchers split training set, validation set and test set from the same dataset. It will be interesting to know whether the relation extraction model could still function well as expected, even if it was only trained in much simpler training set. If it is possible, this would give the chance for future researchers to have a lightweight training while guaranteeing quality results.

When we test a model in test set and get result X, we never know this is caused by which part of the model or if there is unexpected magic between some parts of the model. What is the relation between the length of sentences and performance of relation extraction, how the size and complexity of training set will affect relation extraction and how length of documents will interact with other components of a relation extraction model? These answers are still unclear and will be addressed in experiments.

Aspect 4: Embeddings

The forth observation is that current neural network based methods are taking different word embeddings as input. It is the widely used way of mapping natural languages into high dimensions, so that each word has a coordinate in vector space which represents its syntax

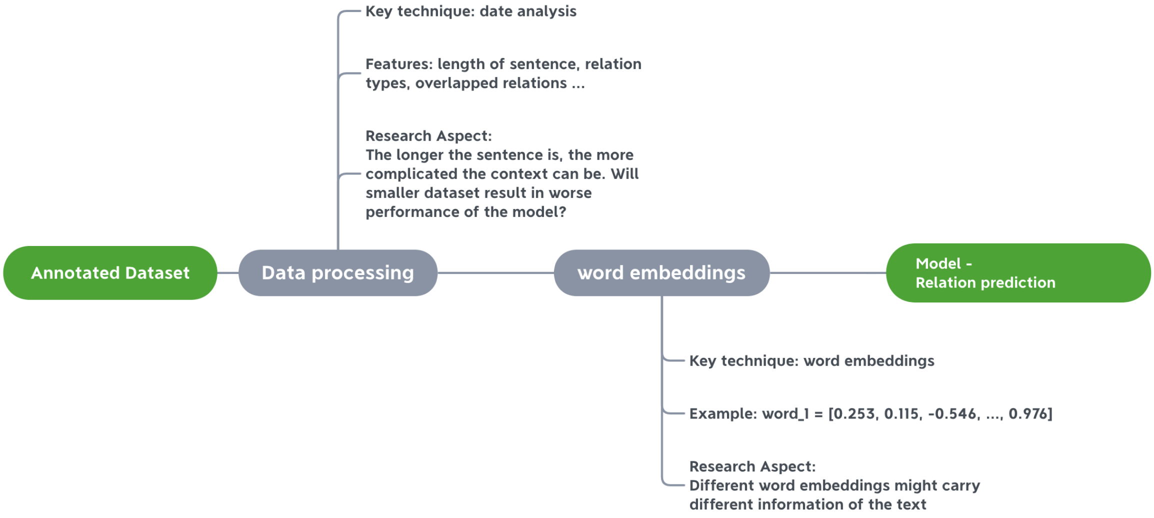

Key technique: date analysis
Features: length of sentence, relation types, overlapped relations ...
Research Aspect: The longer the sentence is, the more complicated the context can be. Will smaller dataset result in worse performance of the model?
Annotated Dataset
Data processing
word embeddings
Model - Relation prediction
Key technique: word embeddings
Example: word_1 = [0.253, 0.115, -0.546, ..., 0.976]
Research Aspect: Different word embeddings might carry different information of the text

Figure 3.12: Experiment Design Part 2

meaning. The words with similar meaning will be located closer in the high dimentional space. The classic pre-training word embeddings are based on the logic of 'one word, one vector'. This means the embedding is fixed no matter what context the word is in. Popular examples of classic word embeddings are *Word2Vec*[30], *FASTTEXT*[40] and *GloVe*(Global Vectors for Word Representation)[65]. These are also regarded as static word embeddings.

Recent work has been focused on different word embeddings, also known as dynamic word embeddings, for the same word in various contextual background. Examples like *Bert Embeddings*(Bidirectional Encoder Representations from Transformers)[20], *ELMo Embeddings*(Embeddings from Language Models)[68] and *Flair Embeddings*[4, 3] are contextualized.

Compared to static word embeddings, dynamic ones are capable of understanding polysemy. Also, classic word embeddings do not consider the order of words while contextual word embeddings takes the order as information, thanks for the usage of LSTM in *ELMo Embeddings* and Transformer in *Bert Embeddings*. These frameworks showed promising results on various NLP tasks in the NLP fields of NER and part-of-speech (PoS) tagging. The current state-of-the-art is *Flair Embeddings*[5], which used a sequence labeling architecture with the foundation of neural language modeling. The framework of *Flair Embeddings* is referenced in Figure 3.13.

As shown with the directions of arrows, *Flair Embeddings* concatenates information both from the beginning of the sentence till the last character of the target word and from the end of the sentence till the first character of the word. By doing this, the final embedding carries the global contextual meanings.

In the benchmark model, the pre-trained static word embeddings were used to represent syntax meanings. In this thesis, we decided to take dynamic word embeddings as the

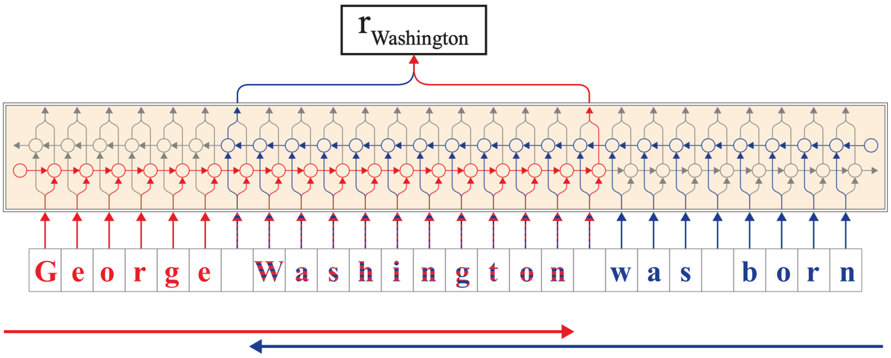


Figure 3.13: The Framework of Flair Embeddings[5]

input for the model, considering that it is more aligned with reality to have different meanings for words with different contextual background information. We made use of *Flair* framework to generate *Bert Embeddings*, *ELMo Embeddings* and *Flair Embeddings*. However, although we tried to reduce the dimensions of word embeddings from 4096 into 2048, the whole word embeddings file already occupied more than 200GB. And the final word embeddings was shaped as 2359644 * 2048 dimensions, which took around two weeks to generate. This resulted in the failure of loading the whole embeddings and feeding into the bidirectional RNN encoder of the benchmark model, due the limited GPU memory. We also investigated the possibility of chunking the embeddings and read them one by one. Unfortunately, this was not supported by the used *TensorFlow*. After months of trying, we had to give up the part of word embeddings in this project due to limited resources. We are convinced that word embeddings which consider overall contextual information can greatly benefit machine in NLU. This will be continued as future work once the resources requirements are met.

Chapter 4

Experiment

In this chapter we will elaborate on the research directions we have set out in Chapter 2 and Chapter 3 through further investigation and scientific experiments. The experiments can be outline in the following stages:

- **Data Preprocessing and Analysis**
  In this section we will explain how the selected dataset were processed in order to create a training set, validation set and test set. Furthermore, characteristics of each dataset, such as the number of tokens, the distribution of length of sentences, related entity and relation types, will be described.
- **Evaluation Metric**
  The section mainly focuses on the evaluation metrics we use for comparing methodologies, components and algorithms in this thesis.
- **Result Analysis**
  In this section, we will visualize the results and analyze them as seen from different perspectives. The analysis will be split according to various experiment aspects we illustrate in the previous chapters.
- **Summary**
  Lastly, the defined research questions will be answered in detail, supported by the experiment results. After reading this section, the reader is able to have a overview of the contributions of this thesis.

4.1 Data Preprocessing and Analysis

Based on the summary on Chapter 2 and Chapter 3, we use the same texts for training as Zeng et al. [107] did. Zeng et al. [107] pre-processed the training set from Ren et al. [74] by excluding documents, which are longer than 100 words and/or only include 'None' relations. After the step of data cleaning, there are 66196 documents left.

If we regard this as an example document with 9 word tokens: *Stefan Zweig*[Person] studied philosophy at the *University of Vienna[Location]*[Organization], there are 3 entity instances from entity instance perspectives ([Person], [Location] and [Organization]). And *Stefan Zweig* and *University of Vienna* are relation instances of the relation *graduate from*. Moreover, we define [*Stefan Zweig*, *University of Vienna*, *graduate from*] as a triple. Based on these definitions, the statistics are elaborated in Table 4.1. Additionally, we name the original training set from Zeng et al. [107] as training set 1 so that later modifications of the training set will not be mixed.

Dataset	Documents	Entities Instances	Entities/ Document	Relations Instances	Triple Set	Tokens	Average Tokens
Training Set 1	66196	209245	3.16	111327	17621	2503189	37.81

Table 4.1: The Statistics of Original Training Set

In total, the original training set contains 24 positive relation types and a negative relation (None). Details can be found in Table 4.2.

Relation Type	Number	Relation Type	Number
/location/location/contains	53699	/people/person/nationality	8430
/location/country/capital	8042	/people/person/place_lived	7513
/location/country/administrative_divisions	6796	/location/administrative_division/country	6796
/business/person/company	5852	/location/neighborhood/neighborhood_of	5804
/people/person/place_of_birth	3311	/people/deceased_person/place_of_death	2021
/business/company/founders	836	/people/person/children	529
/business/company/place_founded	433	/business/company_shareholder/ major_shareholder_of	303
/business/company/major_shareholders	303	/sports/sports_team_location/teams	225
/sports/sports_team/location	225	/people/person/religion	71
/business/company/advisors	47	/people/ethnicity/geographic_distribution	44
/people/ethnicity/people	22	/people/person/ethnicity	22
/people/person/profession	2	/business/company/industry	1

Table 4.2: The Details of Relations in Training Set 1

Using this training set, which is originally created by Ren et al. [74] and cleaned by Zeng et al. [107], we re-implement the experiments of Zeng et al. [107]. There are two highlights worth mentioning.

- Zeng et al. [107] directly pre-processed the dataset generated by Ren et al. [74]. However, it is not annotated by the distant supervision model CoType designed by Ren et al. [74].

- Although using training set from CoType, Zeng et al. [107] did not use the same test dataset as Ren et al. [74]. The test set is split from the processed training set.

In terms of the first observation, the question is that why did not Ren et al. [74] use their own model for annotation of this dataset, even with pages of explanations of their model in the paper? Supported by results of our re-implementation, we believe this is because CoType cannot annotate all predefined relations in the training set based on the relation types the test set. This means there are some labels in test set but those labels are not be able to annotate by the CoType model.

We carefully annotate the same texts using the distant supervision model from Ren et al. [74] and get the dataset (training set 2) with statistics in Table 4.3. By comparing Table 4.2 and 4.3, there is an obvious difference: the number of relation types. In the training set 1, there are 24 positive relation types. But only 14 relation types in training set 2, which is annotated by the distant supervision model. This leads to the fact that if they use their own distant supervision model, the training set will miss many target relations. Consequently, the experimental results would not be ideal. Based on this observation, our guess is that the authors gathered the information from the test set before they chose the model. In order to annotate a dataset with similar distribution as the test set, they chose the annotation model from Riedel et al. [75].

Relation Type	Number	Relation Type	Number
location.location.contains	100880	people.person.nationality	14517
location.country.capital	10510	location.country.administrative_divisions	9373
location.neighborhood.neighborhood_of	8211	people.person.place_of_birth	5267
organization.organization.founders	3352	people.deceased_person.place_of_death	2309
people.person.children	1048	organization.country.place_founded	855
sports.sports_team.location	188	sports.sports_team_location.teams	188
organization.organization.advisors	68	people.person.religion	3

Table 4.3: The Relation Types Annotated by the Distant Supervision Model from Ren et al. [74]

For the second point, Zeng et al. [107] split the training, validation and test sets by themselves for the reason that the test set, used by Ren et al. [74], contains only single relation between a pair of entities. the test dataset used by Ren et al. [74], is annotated manually by crowd sourcing from Hoffmann et al. [35]. This is the most popular test set for distant supervision models. However, the most promising point of the model from Zeng et al. [107] is that it is able to detect more than one relation for a pair of entities. If they still use the same test set as other researchers, the experimental results will not show the advantages of their model. In the test set (marked as test set 1 for clarification) from Zeng et al. [107], there are 395 documents. This is created by randomly sampling from the pre-processed training set 1. After conducting the same pre-process steps on number of tokens and negative labels, there are 392 documents left. Statistics are elaborated in Table 4.4 and 4.5.

Based on these two important observations, we came up with our experimental workflow. As mentioned, in order to build the pipeline from annotation to relation extraction, we

Dataset	Documents	Entities Instances	Entities/ Document	Relations Instances	Triple Set	Tokens	Average Tokens
Training Set 1	392	1346	3.43	407	286	14922	38

Table 4.4: The Statistics of Original Test Set 1

Relation Type	Number	Relation Type	Number
/location/location/contains	178	/location/administrative_division/country	108
/people/person/place_lived	40	/business/person/company	38
/people/person/nationality	27	/business/company/founders	5
/location/country/administrative_divisions	3	/location/country/capital	2
/people/deceased_person/place_of_death	2	/people/person/children	2
/location/neighborhood/neighborhood_of	1	/people/person/place_of_birth	1

Table 4.5: The Details of Relations in Test Set 1

used the distant supervision model from Ren et al. [74] and the relation extraction model from Zeng et al. [107]. We elaborate the reasoning in the previous section. So even if the training set 1 used by Zeng et al. [107] is not annotated by Ren et al. [74] because of extra information from test set, this is out of the scope of this thesis. What we pursue is a promising pipeline and reasonable experiments to compare the importance of different components in relation extraction. Thus, we use the same texts from the training set of Ren et al. [74]. To guarantee consistency, we annotate the texts using the model of Ren et al. [74]. And following Zeng et al. [107], we pre-process the dataset and split the training, validation and test sets from the annotated dataset. This means our training set and test set are different from the benchmark model in Zeng et al. [107].

Comparatively, we create our own training set 2, validation set 2 and test set 2 by annotating dataset with the distant supervision model from Ren et al. [74] and then also randomly sampling and pre-processed the documents as Zeng et al. [107]. The randomly sampled validation set 2 are split into 10 portions during experiment, which were used for evaluation during training. In Figure 4.1 and 4.2, the distribution of tokens after data cleaning is shown.

The statistics of our datasets can be found below.

Dataset	Documents	Entities Instances	Relation Types	Relation Instances	Triple Set	Tokens	Average Tokens
Training Set 2	80074	309516	14	18614	22190	3100077	38.72
Validation Set 2	5000	19366	13	9035	4069	193543	38.71
Test Set 2	395	1560	12	528	553	15669	39.67
In total	85469	330443	14	166074	23216	3309289	38.72

Table 4.6: The Statistics of Training Set 2, Validation Set 2 and test set 2

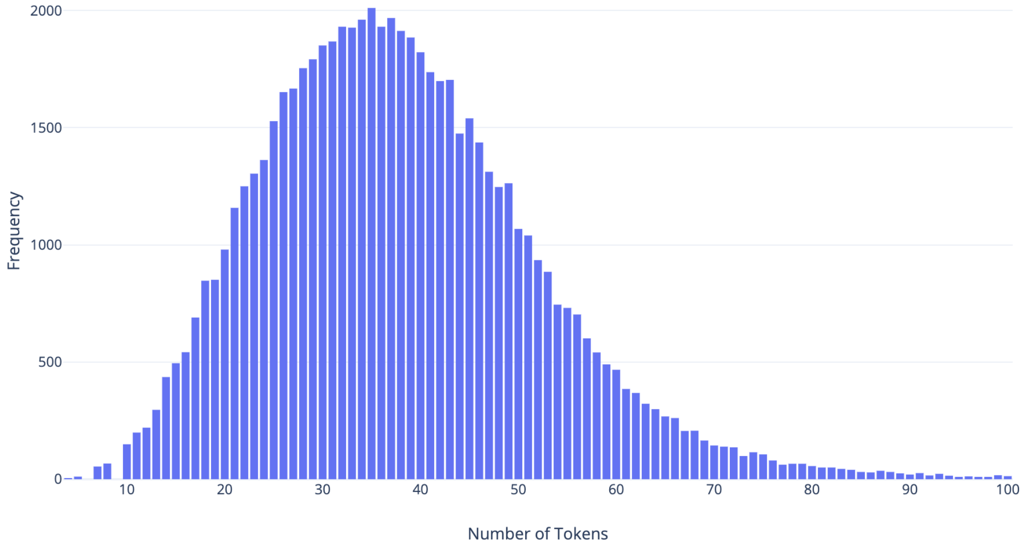


Figure 4.1: The Distribution of Tokens in Training Set 2

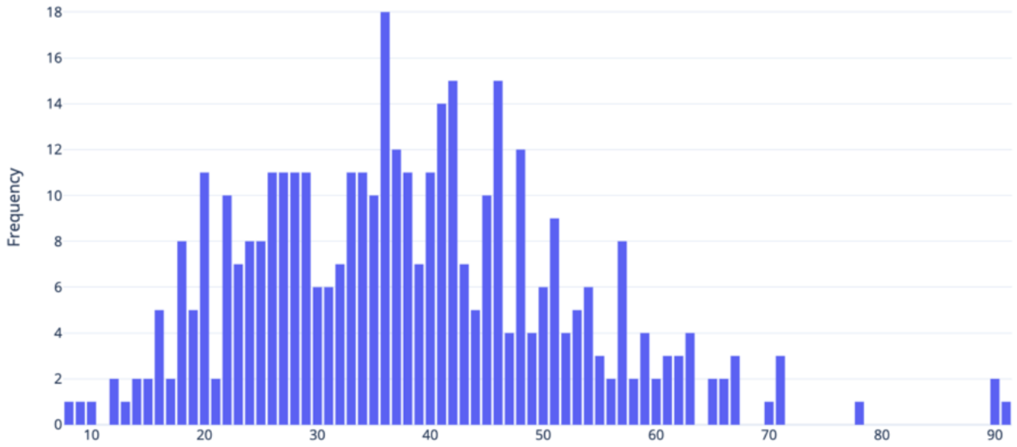


Figure 4.2: The Distribution of Tokens in Test Set 2

Relation Type	**Training**	**Validation**	**Test**	**Relation Type**	**Training**	**Validation**	**Test**
location.location.contains	94631	5761	488	people.person.nationality	13564	894	59
location.country.capital	9854	612	44	location.country.administrative_divisions	8818	519	36
location.neighborhood.neighborhood_of	7700	467	44	people.person.place_of_birth	4948	301	18
organization.organization.founders	3136	189	27	people.deceased_person.place_of_death	2167	135	7
people.person.children	977	70	1	organization.country.place_founded	804	49	2
sports.sports_team_location.teams	168	17	3	sports.sports_team.location	168	17	3
organization.organization.advisors	64	4	0	people.person.religion	3	0	0

Table 4.7: The Relation Types in Training set 2, Validation Set 2 and Test Set 2

4.2 Evaluation Metric

We applied standard Precision, Recall and F1 score to evaluate the results as Zeng et al. [107]. A triplet, for example [Stefan Zweig, University of Vienna, graduate from], is regarded as correct when both the relation type and the two corresponding entities are successfully recognized.

4.3 Result Analysis

NER is the foundation of relation extraction, which targets the challenge of text variants and ambiguities. The powerful NER models should recognize the relevant Named Entity candidates when the language is written in informal way. The length of sentence in dataset is aiming to investigate the influence of noise and also the complexity of context on the performance of the relation extraction model. And the combination of different parameters gives the insights of whether and how it affects the performance of the relation extraction model.

4.3.1 Performance of Benchmark Model

Based on our experimental settings, we use training set 2, which is annotated by Cotype model, as the basic training set of the experiments. Then we extract relations using the model from Zeng et al. [107]. In Figure 4.4 and Table 4.8, we can find the experimental results of benchmark model in validation set 2 and test set 2. Moreover, we list some detected examples by the model based on the categories of normal instance (one relation between a pair of relation instances), multi instance(several relations between a pair of relation instances) and overlapping instance(several relations between three relation instances). The is addressed and aligned with the gap we identified during related work.

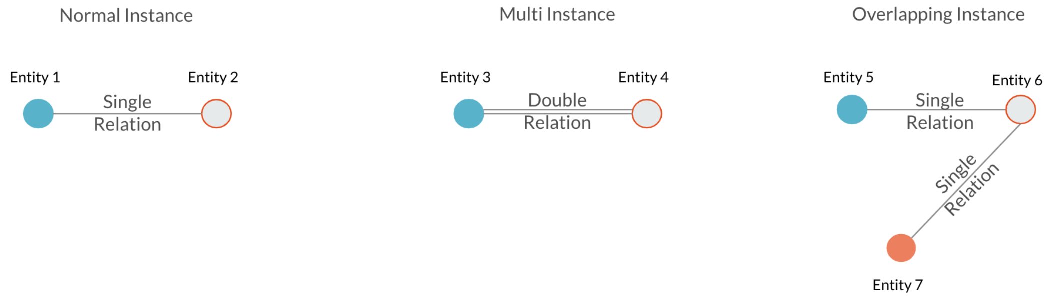


Figure 4.3: The Visualisation of Complicated Relation Instances

Normal instance:
Most surprisingly , 55 percent of people ages 18 to 25 rejected the treaty, underscoring what appeared to be a lack of trust in the future of Europe and the leadership of France.

Gold: [Europe, France, location.location.contains]
Predict: [Europe, France, location.location.contains]

Toyota plans to increase production outside of Japan by 40 percent , to five million vehicles, by 2008 as it tries to top General Motors as the world 's largest automaker.
Gold: [Toyota, Japan, organization.country.place_founded]
Predict: [Toyota, Japan, organization.country.place_founded]

Multi instance:
When the Communists came to power in 1948 , he was smuggled across the border into Austria and resided in Vienna before immigrating to the United States in 1949 .
Gold: [Vienna, Austria, location.country.capital] [Austria, Vienna, location.location.contains] [Austria, Vienna, location.country.administrative_divisions]
Predict: [Vienna, Austria, location.country.capital] [Austria, Vienna, location.location.contains] [Austria, Vienna, location.country.administrative_divisions]

Overlapping instance:
Its flesh-and-blood icons were what he calls , echoing a great old Morrissey song , " The Last of the Famous International Playboys , " or such life-sweepstakes winners as Warren Beatty and Hugh Hefner in America , and Sacha Distel and Jean-Paul Belmondo in France.
Gold: [Belmondo, France, people.person.nationality] [Distel, France, people.person.nationality]
Predict: [Belmondo, France, people.person.nationality] [Distel, France, people.person.nationality]

Kenneth Lay and Jeffrey Skilling , the former chief executives of Enron , had their day in court.
Gold: [Enron, Skilling, organization.organization.founders]
Enron, Lay, organization.organization.founders
Predict: [Enron, Skilling, organization.organization.founders]
Enron, Lay, organization.organization.founders

	Precision	Recall	F1-score
Benchmark performance	0.807	0.728	0.766

Table 4.8: The Performance of Benchmark Model in Test Set 2

4.3.2 Experiments Results of Benchmark Model with NER Models

In this section, we analysed the difference between different NER models. Because distant supervision is working based on the detected NER candidates, it affects the quality of the dataset directly. IN Table 4.9, 4.10 and 4.11, we summarised the statistics of training set while using different NER models. And in Table 4.12, 4.13 and 4.14, we documented the overview of the distribution of different relation types while using different NER models.

Compared to the NER model used in benchmark model (Stanford NER), both Flair and NLTK tend to have fewer documents with valid annotations. Although NLTK NER could

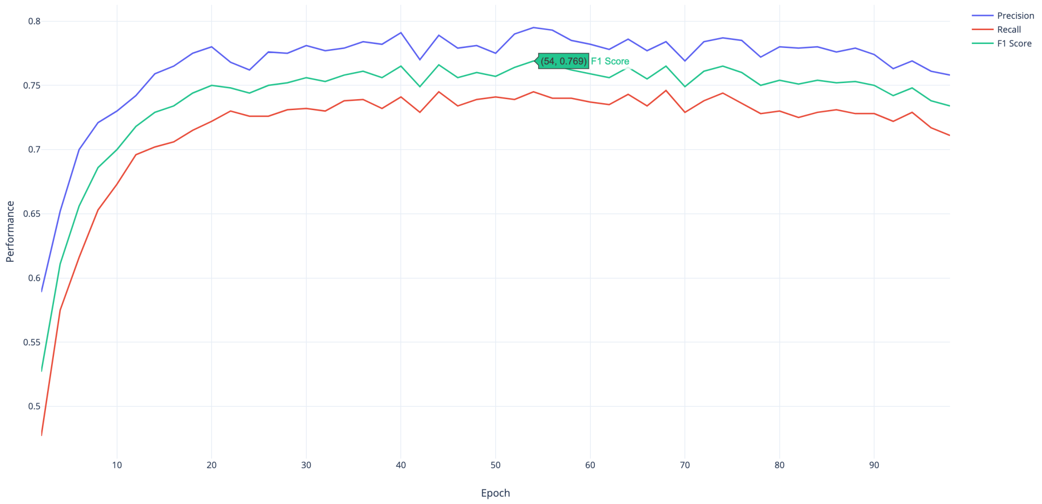


Figure 4.4: The Performance of Original Relation Extraction Model in Validation Set 2

result in the most entity instance in total and per document, the number of valid relation instances the significantly lower than Stanford NER and Flair NER. Another observation is that Flair NER performs in general better in longer documents, supported by the number of tokens, average tokens and number of documents. Comparatively, NLTK works the worst in documents with more tokens. In summary, the differences between various NER models result in various training set with different size, levels of complexity of context and different valid annotations. We used these training set to train the benchmark model and tested in the same test set, which would be elaborated in later section.

Dataset	Documents	Entities Instances	Entities/ Document	Relations Instances	Triple Set	Tokens	Average Tokens
Training Set 2	85469	330443	3.86	166074	23216	3309289	38.72

Table 4.9: The Statistics of the Training Set with Stanford NER

Dataset	Documents	Entities Instances	Entities/ Document	Relations Instances	Triple Set	Tokens	Average Tokens
Training Set 3	66562	342788	5.15	126848	19072	2506486	37.66

Table 4.10: The Statistics of the Training Set with NLTK NER

Dataset	Documents	Entities Instances	Entities/ Document	Relations Instances	Triple Set	Tokens	Average Tokens
Training Set 4	85206	339137	3.98	157766	23322	3386860	39.75

Table 4.11: The Statistics of the Training Set with Flair NER

Regarding annotations of relation types, in general Stanford NER performed similar as Flair NER, while NLTK NER annotated much fewer relation instances across all relation types. This is reasonable considering that NLTK NER failed to annotate about 20000 documents compared to other NER models.

Relation Type	Number	Relation Type	Number
location.location.contains	100880	people.person.nationality	14517
location.country.capital	10510	location.country.administrative_divisions	9373
location.neighborhood.neighborhood_of	8211	people.person.place_of_birth	5267
organization.organization.founders	3352	people.deceased_person.place_of_death	2309
people.person.children	1048	organization.organization.place_founded	855
sports.sports_team.location	188	sports.sports_team_location.teams	188
organization.organization.advisors	68	people.person.religion	3

Table 4.12: The Relation Types Annotated by Stanford NER

Relation Type	Number	Relation Type	Number
location.location.contains	80322	people.person.nationality	8918
location.country.capital	8399	location.administrative_division.country	7326
ocation.country.administrative_divisions	7326	location.neighborhood.neighborhood_of	6380
people.person.place_of_birth	3092	organization.organization.founders	2055
people.deceased_person.place_of_death	1495	people.person.children	644
organization.organization.place_founded	670	sports.sports_team.location	77
sports.sports_team_location.teams	77	people.person.profession	65
people.person.religion	2		

Table 4.13: The Relation Types Annotated by NLTK NER

Relation Type	Number	Relation Type	Number
location.location.contains	101867	people.person.nationality	14459
location.country.capital	10375	location.country.administrative_divisions	9260
location.neighborhood.neighborhood_of	8514	people.person.place_of_birth	5186
organization.organization.founders	3455	people.deceased_person.place_of_death	2299
people.person.children	1045	organization.organization.place_founded	851
sports.sports_team.location	190	sports.sports_team_location.teams	190
organization.organization.advisors	68	people.person.religion	7

Table 4.14: The Relation Types Annotated by Flair NER

In Figure 4.4, 4.5 and 4.6, we visualised the performance of relation extraction in validation set with different NER models. And in Table 4.15, we could see the performance of them in test set, by Precision, Recall and F1-Score. The model with NLTK NER has the best F1-Score in validation set but worked the worst in test set. This is easy to understand considering that test set includes documents with more tokens but NLTK NER annotated relatively shorter documents. It performed worse than benchmark model with Stanford NER from all evaluation metrics. However, the model with Flair NER outperformed the bench-

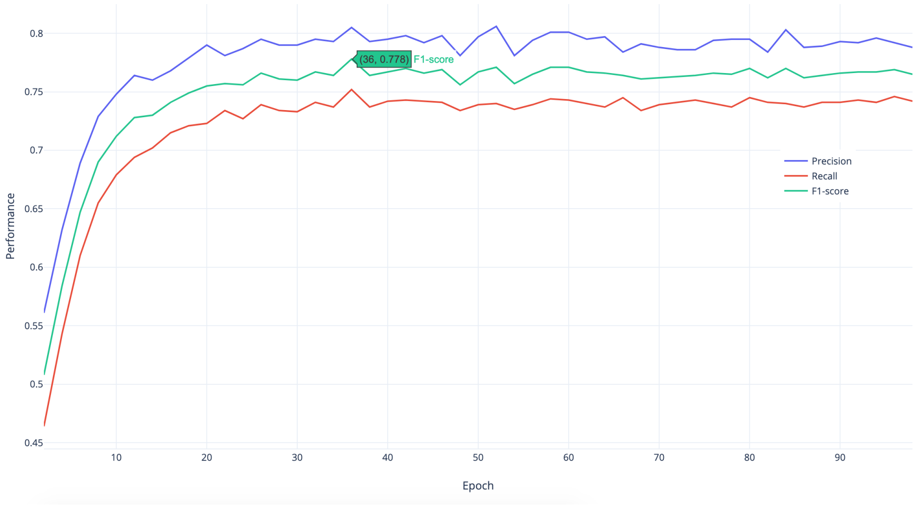


Figure 4.5: The Performance in Validation Set of Relation Extraction Model with NLTK NER

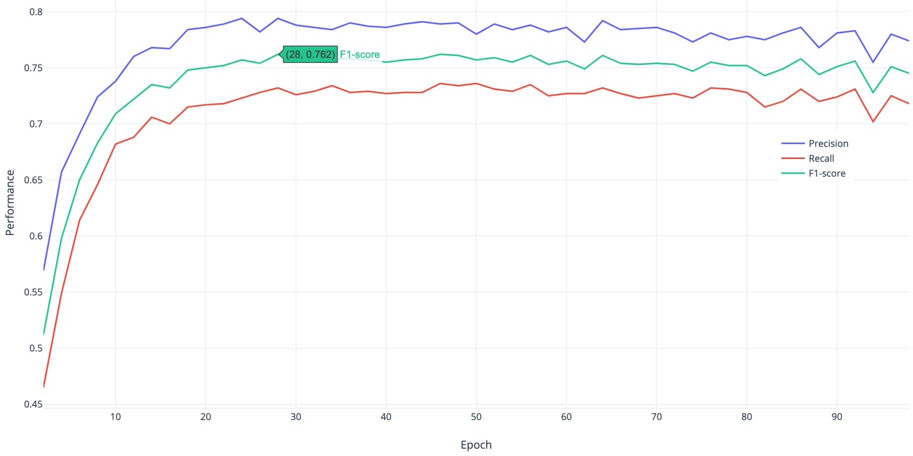


Figure 4.6: The Performance in Validation Set of the Relation Extraction Model with Flair NER

mark model by almost 10% in F1-Score. Its Precision and Recall were also promising. For all models with various NERs, the Precision is generally higher than Recall.

In general, these results are interesting for the reason that these are aligned with our analysis of statistics of annotations in the previous section regarding various NER models. The NER model, which worked best in identifying relation instances in documents with longer sentences, significantly improved the performances of relation extraction.

NER	Best-performing Epoch in Validation	Best Validation Result	Precision	Recall	F1-Score
Stanford	54	0.769	0.807	0.728	0.766
NLTK	36	0778	0.737	0.657	0.695
Flair	28	0.762	0.881	0.831	0.855

Table 4.15: The Performance of the Benchmark Model with Different NER in Test Set 2

4.3.3 Experiments Results of the Benchmark Model with Different Length of Documents in Training Set

In this section, we regarded the maximum length of documents in training set as the experimental aspect. In Figure 4.7, we could see the distribution of number of tokens in training set. The majority of the documents has around 30 to 50 tokens. By limiting the maximum length of documents, we were able to analyse how robust the model of relation extraction was and how dependent it was on the complexity of contexts. Shorter documents tend to have fewer complicated relation instances like multi instances and overlapping instances, which were defined at the beginning of this chapter. The benchmark model has documents maximally with 100 tokens, as same as test set. Because the target of these experiments is the maximum length of documents, we kept other components of the benchmark as it was.

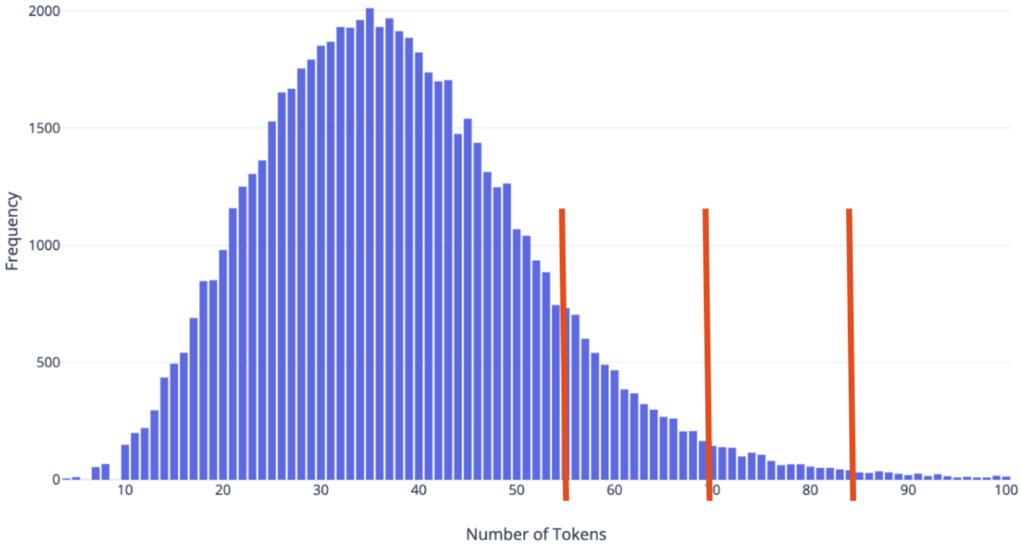


Figure 4.7: The Visualisation of Different Maximum Length of Documents in Training set

In Figure 4.8, 4.9 and 4.10, we could see the visualisation of performances of the benchmark model with different maximum length of documents in validation set. And in Table 4.16, we provided an overview of the performances across various length of documents in test set. The overall trend is that the model performed worse as the documents were getting

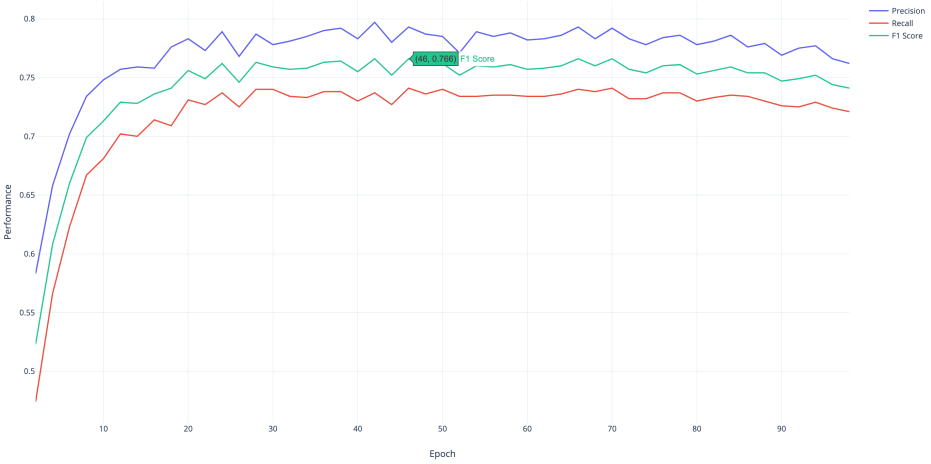


Figure 4.8: The Performance in Validation Set of the Relation Extraction Model with Max 85 words in Training set

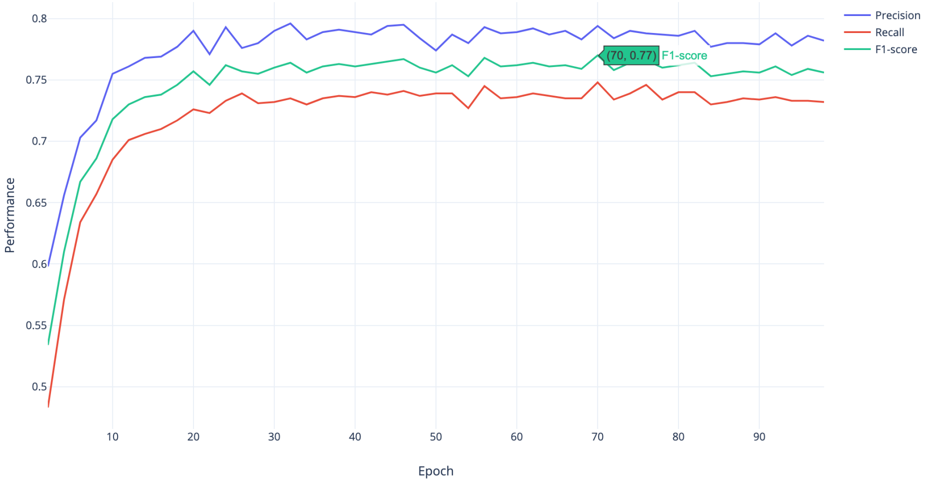


Figure 4.9: The Performance in Validation Set of Relation Extraction Model with Max 70 words in Training set

shorter in training set. It affected Recall the most (dropped by 18.5% when the training set was shortened from 100 tokens to 55 tokens), followed by F1-Score (14.6%).

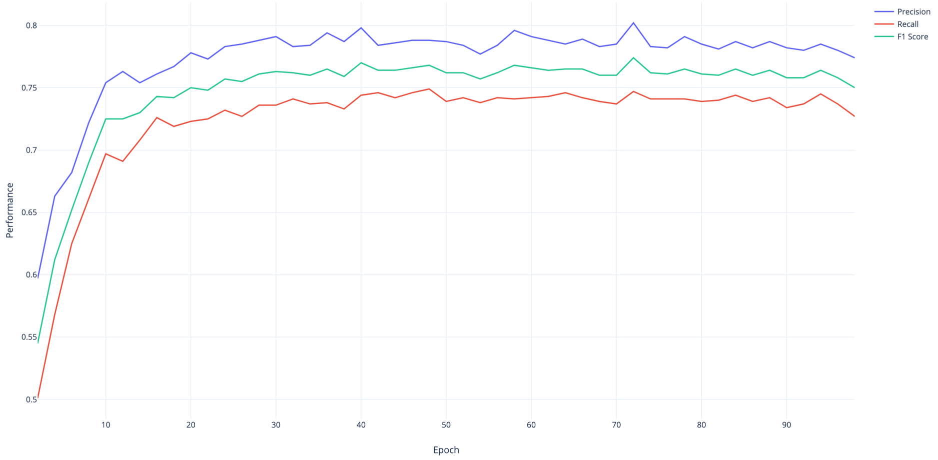


Figure 4.10: The Performance in Validation Set of the Relation Extraction Model with Max 55 words in Training set

Max Length of documents (No. of words)	**Best-performing Epoch during Validation**	**Precision**	**Recall**	**F1-Score**
100	54	0.807	0.728	0.766
85	46	0.775	0.710	0.741
70	70	0.782	0.697	0.737
55	72	0.721	0.543	0.620

Table 4.16: The Performance of the Benchmark Model with Different Length of Documents in Training Set

4.3.4 Experiments Results of Combinations of Different Components

In this section, we combined different parameters of NER models and maximum length of documents and cross analysed the results. This means we experimented how the model of relation extraction performed with Stanford NER, NLTK NER and Flair NER, combining maximum length of documents in training set from 100 to 55 tokens. In Figures below, we again visualised the performance in validation set of relation extraction model in different settings. And in Table 4.17, we summarised all test results. Four settings from maximum length of documents and three possibilities from NER models made 12 experimental results in total.

Regarding the extent of the decline, the F1-score of the model with Flair NER dropped 13.6% when the max length of documents shortened from 100 into 55. The corresponding statistics of the models for Stanford NER and NLTK NER were the decrease of 14.6% and the increase of 4.1%. This showed that benchmark model with Stanford NER was the most heavily influenced by the complexity of context in training set among all NER models.

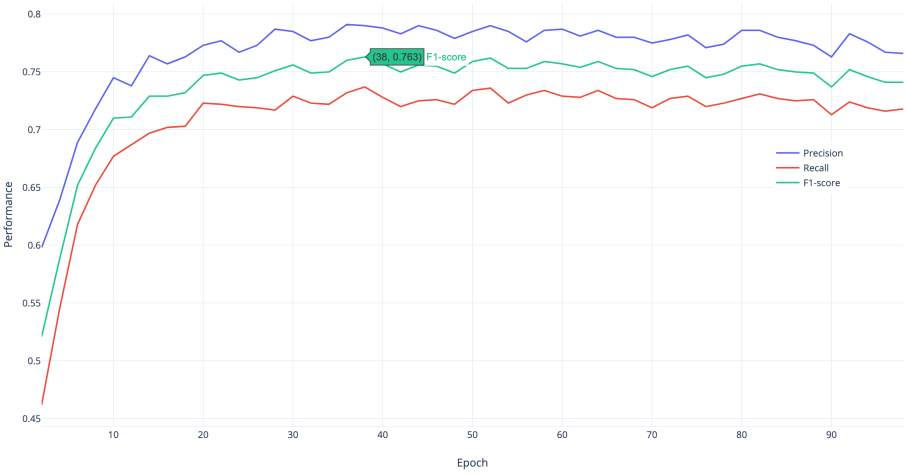


Figure 4.11: The Performance in Validation Set of the Relation Extraction Model with Flair NER and Max 85 words in Training Set

From F1-Score point of view, the model of relation extraction with Flair NER outperformed other models, no matter what was the maximum length of documents. The strongly suggested that the model of relation extraction with Flair NER was robust and suitable for datasets regardless of the complexity of context. In contract, the model with NLTK NER had the trend of working better with smaller maximum length of documents. The benchmark model with Stanford NER performed better than the model with NLTK NER in most cases, except when the maximum length of sentence was 55.

We made the visualisation to present the results in a better way, which could be found in Figure 4.14. We highlighted three observations. First of all, although the model with Flair NER worked the best generally, it with NLTK performed basically as well as the model with Flair NER when the maximum length of documents was limited as 55 words. Sec-

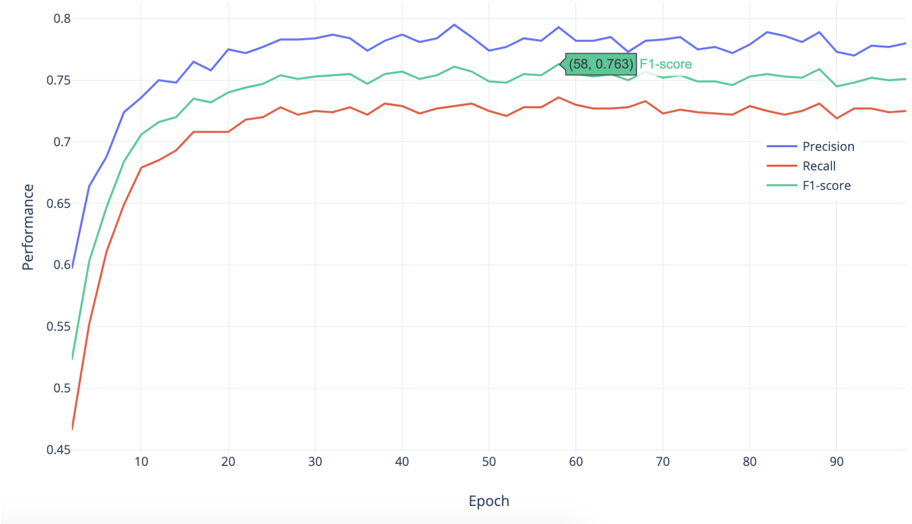


Figure 4.12: The Performance in Validation Set of the Relation Extraction Model with Flair NER and Max 70 words in Training Set

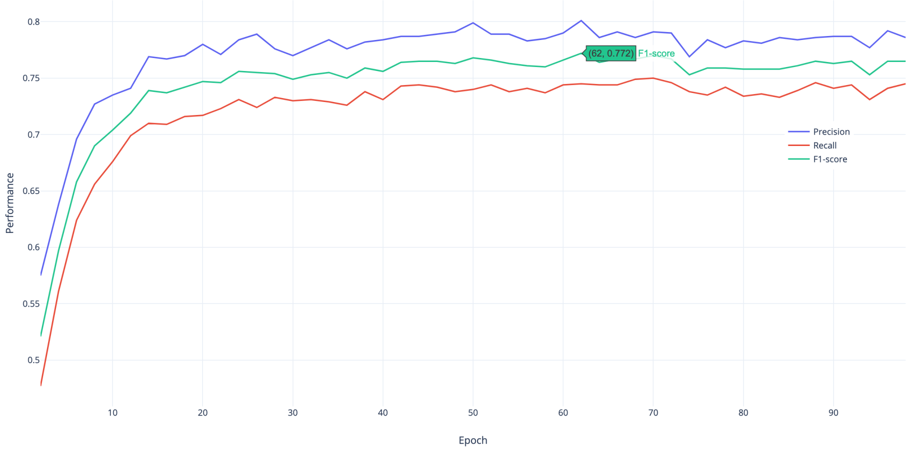


Figure 4.13: The Performance in Validation Set of the Relation Extraction Model with Flair NER and Max 55 words in Training Set

ondly, when the maximum length of documents was shorter than about 65, the model with NLTK functioned better than the benchmark model with Stanford NER. Lastly, when we used NLTK NER, it was the only case where there was the decreasing trend with gradually increasing length of documents.

In terms of Recall, it appeared to be similar trend as F1-score, except when the maximum length of documents was 55. The model with NLTK NER surprisingly performed better than all other models in this case. This again addressed the preference of relation

Max Length of documents (No. of words)/NER	Best-performing Epoch in Validation	Best Validation Result	Precision	Recall	F1-Score
100/Stanford	54	0.769	0.807	0.728	0.766
100/Flair	28	0.762	0.881	0.831	0.855
100/NLTK	80	0.768	0.722	0.633	0.675
85/Stanford	46	0.766	0.775	0.710	0.741
85/Flair	38	0.763	0.851	0.805	0.828
85/NLTK	38	0.773	0.730	0.652	0.688
70/Stanford	70	0.770	0.782	0.697	0.737
70/Flair	58	0.763	0.846	0.772	0.807
70/NLTK	82	0.775	0.716	0.646	0.679
55/Stanford	72	0.774	0.721	0.543	0.620
55/Flair	62	0.772	0.796	0.657	0.720
55/NLTK	36	0.776	0.744	0.690	0.716

Table 4.17: The Performance of the Model with Different NER and Different Length of Documents in Training Set

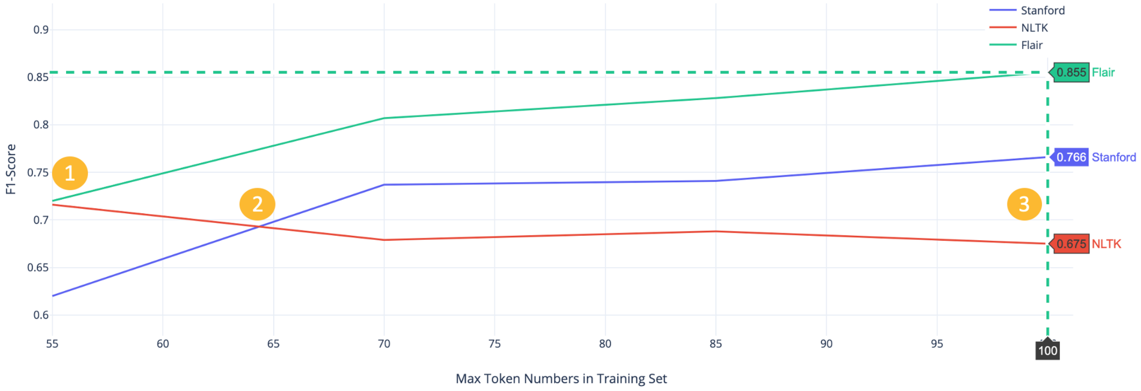


Figure 4.14: The F1-Score in Test Set with the Combination of Different NERs and Different Max Length of Sentence in Training set

extraction model with NLTK NER to be shorter and simpler documents. The Recall of the model with Stanford NER decreased by 18.5%, followed by 17.4% from the model with Flair NER. The model with NLTK NER increased by 5.7% while the longest documents was shortened from 100 tokens to 55 tokens. As visualised in Figure 4.15, the usage of NLTK NER resulted in the best Recall when we removed all documents which contained more than 55 words. It started showing an obvious climbing trend after the documents got shorter than 70 tokens. In general, the application of Flair NER still stood out in most cases regarding Recall.

Precision shared the similar observations as other evaluation metrics. However, there were a few exceptions. On the one hand, the model with NLTK NER only showed very minor difference comparing to the benchmark model. And its out-performance happened only when the maximum length of documents were close to 55. This showed that the model tended to label irrelevant texts with predefined annotations of relation instances. The

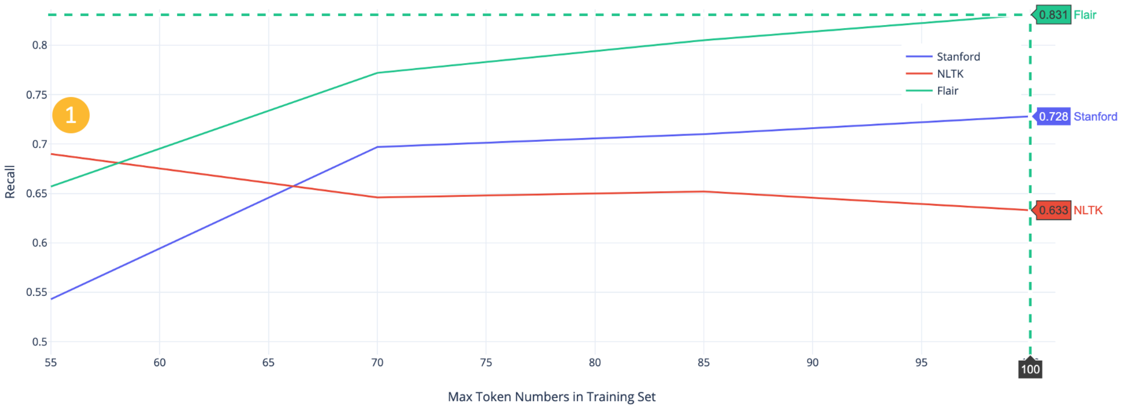


Figure 4.15: Recall in Test Set with the Combination of Different NERs and Different Max Length of Sentence in Training set

difference of Precision across models is the minimal (15.9%), compared to its of Recall (19.8%) and F1-Score (18%). The biggest difference always appeared when the maximum length of documents is the longest (100 tokens).

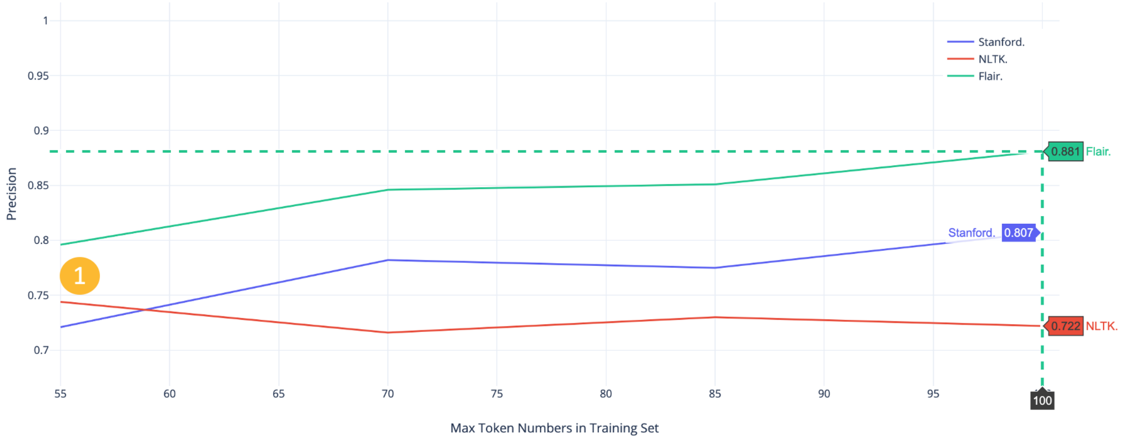


Figure 4.16: Precision in Test Set with the Combination of Different NERs and Different Max Length of Sentence in Training set

Summary

The experimental results highlighted some key observations. First of all, by investigating the insights of the model instead of changing the framework, the performance of relation extraction can be dramatically improved. This conveys the message that the framework of the model is not the only decisive factor. The dependencies and insights of components worth more attention. Without investigation of insights of the model, it is difficult to tell whether the performance stands out because of its capability or the biased perspective.

Secondly, a powerful and accurate Named Entity Recognizer is capable of significantly increasing F1-Score by 8.9% in our case. The performance of relation extraction is overall aligned with the analysis across various models of Named Entity Recognition. The Named Entity Recognition model, which works best in identifying relation instances in documents with longer sentences, can improve the performances of relation extraction. This addresses the importance of Named Entity Recognition in the task of relation extraction.

Lastly, it is proven that a model can achieve better results even if the similarity between training set and test set is weaker. The standard way of splitting training, validation and testing set does not always guarantee the quality performance in testing. Figuring out the preferences of components in the relation extraction can benefit the result (4.1% improve of F1-Score in our experiment). This means it is possible that we train the models in a much lighter training set and achieve quality results in testing.

Chapter 5

Discussion

In this chapter we summarize the contributions of this thesis in relation to the research questions stated in Chapter Introduction. We then describe future research directions.

5.1 Conclusion

In the Introduction chapter we formulate the following research question. Hereby, we explain how we answered the research question.

- **The Main Question:**
  How to improve the performance of relation extraction?
  Supported by our experiments, investigating the insights of components without changing the framework of the model, the performance of relation extraction can be dramatically improved.

- **Subquestion 1:**
  Except the model itself, what are influential components in a relation extraction pipeline?
  Influential components at least include Named Entity Recognition and maximum length of documents in training set, supported by our experiments. Entity Linking and word embeddings are very promising based on our literature review.

- **Subquestion 2:**
  How can these components affect the performance of relation extraction?
  A strong Named Entity Recognizer has the capability of increasing F1-Score by 8.9%. With a proper combination of Named Entity Recognition and maximum length of documents in training set, the worst performing model can achieve obviously better results in some cases. Also, it is proven that promising test results can be achieved though we train the model in a smaller and simpler training set.

To answer the research questions, we proposed a pipeline of complex relation extraction from unlabeled plain text based on distant supervision and sequence-to-sequence learning with copy mechanism. We investigated and identified influential components within the pipeline, namely Named Entity Recognition, Entity Linking, maximum length of documents in training set and word embeddings. We conducted cross experiments by applying different models of each component and tuning parameters to analyze influence of the specific components on the performance of the relation extraction model.

5.2 Future work

In the future, more components will be evaluated and analysed, which include at least Entity Linking and word embeddings. We have the ambition of visualising how performances of relation extraction will fluctuate when the commonly shared components across models of relation extraction are tuned. This will guide the current and future researches to improve from where they are instead of adapting another framework.

Bibliography

[1] Gustavo Aguilar, Adrian Pastor Lopez Monroy, Fabio González, and Thamar Solorio. Modeling noisiness to recognize named entities using multitask neural networks on social media. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1401–1412, 2018.

[2] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649, 2018.

[3] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, 2018.

[4] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. Flair: An easy-to-use framework for state-of-the-art nlp. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, 2019.

[5] Alan Akbik, Tanja Bergmann, and Roland Vollgraf. Pooled contextualized embeddings for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 724–728, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1078. URL `https://www.aclweb.org/anthology/N19-1078`.

[6] Alan Akbik, Tanja Bergmann, and Roland Vollgraf. Pooled contextualized embeddings for named entity recognition. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, page to appear, 2019.

[7] Gabor Angeli, Julie Tibshirani, Jean Wu, and Christopher D Manning. Combining distant and partial supervision for relation extraction. In *Proceedings of the 2014*

conference on empirical methods in natural language processing (EMNLP), pages 1556–1567, 2014.

[8] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[9] Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open information extraction from the web. In *IJCAI*, volume 7, pages 2670–2676, 2007.

[10] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[11] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM, 2008.

[12] Sergey Brin. Extracting patterns and relations from the world wide web. In *International Workshop on The World Wide Web and Databases*, pages 172–183. Springer, 1998.

[13] Andrew Chisholm and Ben Hachey. Entity disambiguation with web links. *Transactions of the Association for Computational Linguistics*, 3:145–156, 2015.

[14] Jason PC Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370, 2016.

[15] Yejin Choi, Eric Breck, and Claire Cardie. Joint extraction of entities and relations for opinion recognition. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 431–439. Association for Computational Linguistics, 2006.

[16] Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc V Le. Semi-supervised sequence modeling with cross-view training. *arXiv preprint arXiv:1809.08370*, 2018.

[17] Michael Collins and Brian Roark. Incremental parsing with the perceptron algorithm. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. doi: 10.3115/1218955.1218970. URL https://doi.org/10.3115/1218955.1218970.

[18] Luciano Del Corro and Rainer Gemulla. Clausie: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web*, pages 355–366. ACM, 2013.

[19] Li Deng, Dong Yu, et al. Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387, 2014.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[21] Jason Eisner and Damianos Karakos. Bootstrapping without the boot. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 395–402. Association for Computational Linguistics, 2005.

[22] Yotam Eshel, Noam Cohen, Kira Radinsky, Shaul Markovitch, Ikuya Yamada, and Omer Levy. Named entity disambiguation for noisy text. *arXiv preprint arXiv:1706.09147*, 2017.

[23] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1535–1545. Association for Computational Linguistics, 2011.

[24] Miao Fan, Deli Zhao, Qiang Zhou, Zhiyuan Liu, Thomas Fang Zheng, and Edward Y Chang. Distant supervision for relation extraction with matrix completion. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 839–849, 2014.

[25] Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. Reinforcement learning for relation classification from noisy data. In *Proceedings of AAAI*, 2018.

[26] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

[27] Octavian-Eugen Ganea, Marina Ganea, Aurelien Lucchi, Carsten Eickhoff, and Thomas Hofmann. Probabilistic bag-of-hyperlinks model for entity linking. In *Proceedings of the 25th International Conference on World Wide Web*, pages 927–938. International World Wide Web Conferences Steering Committee, 2016.

[28] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. 1999.

[29] Jeremy Getman, Joe Ellis, Stephanie Strassel, Zhiyi Song, and Jennifer Tracey. Laying the groundwork for knowledge base population: Nine years of linguistic resources for tac kbp. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, 2018.

[30] Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[31] Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2537–2547. The COLING 2016 Organizing Committee, 2016. URL `http://aclweb.org/anthology/C16-1239`.

[32] Zhengqiu He, Wenliang Chen, Zhenghua Li, Meishan Zhang, Wei Zhang, and Min Zhang. See: Syntax-aware entity embedding for neural relation extraction. *arXiv preprint arXiv:1801.03603*, 2018.

[33] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 94–99. Association for Computational Linguistics, 2009.

[34] Lynette Hirschman and Robert Gaizauskas. Natural language question answering: the view from here. *natural language engineering*, 7(4):275–300, 2001.

[35] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 541–550. Association for Computational Linguistics, 2011.

[36] Guoliang Ji, Kang Liu, Shizhu He, Jun Zhao, et al. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *AAAI*, pages 3060–3066, 2017.

[37] Heng Ji and Ralph Grishman. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 1148–1158. Association for Computational Linguistics, 2011.

[38] Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. Overview of the tac 2010 knowledge base population track. In *Third Text Analysis Conference (TAC 2010)*, volume 3, pages 3–3, 2010.

[39] Heng Ji, Joel Nothman, H Trang Dang, and Sydney Informatics Hub. Overview of tac-kbp2016 tri-lingual edl and its impact on end-to-end cold-start kbp. *Proceedings of TAC*, 2016.

[40] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.

[41] Tetsuo Kiso, Masashi Shimbo, Mamoru Komachi, and Yuji Matsumoto. Hits-based seed selection and stop list construction for bootstrapping. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 30–36. Association for Computational Linguistics, 2011.

[42] Natalia Konstantinova. Review of relation extraction methods: What is new out there? In *International Conference on Analysis of Images, Social Networks and Texts_x000D_*, pages 15–28. Springer, 2014.

[43] Zornitsa Kozareva and Eduard Hovy. Not all seeds are equal: Measuring the quality of text mining seeds. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 618–626. Association for Computational Linguistics, 2010.

[44] Kevin Lange Di Cesare, Amal Zouaq, Michel Gagnon, and Ludovic Jean-Louis. A machine learning filter for the slot filling task. *Information*, 9(6), 2018. ISSN 2078-2489. doi: 10.3390/info9060133. URL `http://www.mdpi.com/2078-2489/9/6/133`.

[45] Fei Li, Meishan Zhang, Guohong Fu, and Donghong Ji. A neural joint model for entity and relation extraction from biomedical text. *BMC Bioinformatics*, 18(1): 198, Mar 2017. ISSN 1471-2105. doi: 10.1186/s12859-017-1609-9. URL `https://doi.org/10.1186/s12859-017-1609-9`.

[46] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *arXiv preprint arXiv:1812.09449*, 2018.

[47] Qi Li and Heng Ji. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412. Association for Computational Linguistics, 2014. doi: 10.3115/v1/P14-1038. URL `http://aclweb.org/anthology/P14-1038`.

[48] Qi Li and Heng Ji. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 402–412, 2014.

[49] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, volume 15, pages 2181–2187, 2015.

[50] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2124–2133, 2016.

[51] Xiao Ling and Daniel S Weld. Fine-grained entity recognition. In *AAAI*, volume 12, pages 94–100, 2012.

[52] Edward Loper and Steven Bird. NLTK: the natural language toolkit. *CoRR*, cs.CL/0205028, 2002. URL `https://arxiv.org/abs/cs/0205028`.

[53] Klaus Macherey, Franz Josef Och, and Hermann Ney. Natural language understanding using statistical machine translation. In *Seventh European Conference on Speech Communication and Technology*, 2001.

[54] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

[55] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[56] Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 777–782, 2013.

[57] Mike Mintz, Steven Bills, Rion Snow, and Dan 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-Volume 2*, pages 1003–1011. Association for Computational Linguistics, 2009.

[58] Makoto Miwa and Mohit Bansal. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770*, 2016.

[59] Makoto Miwa and Yutaka Sasaki. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1858–1869, 2014.

[60] Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244, 2014.

[61] Diego Moussallem, Ricardo Usbeck, Michael Röeder, and Axel-Cyrille Ngonga Ngomo. Mag: A multilingual, knowledge-base agnostic and deterministic entity linking approach. In *Proceedings of the Knowledge Capture Conference*, page 9. ACM, 2017.

[62] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data*, pages 19–34. ACM, 2018.

[63] Axel-Cyrille Ngonga Ngomo, Michael Röder, Diego Moussallem, Ricardo Usbeck, and René Speck. Bengal: An automatic benchmark generator for entity recognition and linking. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 339–349, 2018.

[64] Patrick Pantel and Marco Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 113–120. Association for Computational Linguistics, 2006.

[65] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[66] Maria Pershina, Bonan Min, Wei Xu, and Ralph Grishman. Infusion of labeled data into distant supervision for relation extraction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 732–738, 2014.

[67] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[68] Matthew E Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. Dissecting contextual word embeddings: Architecture and representation. *arXiv preprint arXiv:1808.08949*, 2018.

[69] Minh C Phan, Aixin Sun, Yi Tay, Jialong Han, and Chenliang Li. Pair-linking for collective entity disambiguation: Two could be better than all. *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[70] Van-Thuy Phi, Joan Santoso, Masashi Shimbo, and Yuji Matsumoto. Ranking-based automatic seed selection and noise reduction for weakly supervised relation extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 89–95, 2018.

[71] Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Björne, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski. Bioinfer: a corpus for information extraction in the biomedical domain. *BMC bioinformatics*, 8(1):50, 2007.

[72] Pengda Qin, Weiran Xu, and William Yang Wang. Robust distant supervision relation extraction via deep reinforcement learning. *arXiv preprint arXiv:1805.09927*, 2018.

[73] Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 41–47. Association for Computational Linguistics, 2002.

[74] Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, Tarek F Abdelzaher, and Jiawei Han. Cotype: Joint extraction of typed entities and relations with knowledge bases. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1015–1024. International World Wide Web Conferences Steering Committee, 2017.

[75] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer, 2010.

[76] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84, 2013.

[77] Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. Gerbil–benchmarking named entity recognition and linking consistently. *Semantic Web*, (Preprint):1–21, 2017.

[78] Henry Rosales-Méndez, Aidan Hogan, and Barbara Poblete. Nifify: Towards better quality entity linking datasets. 2019.

[79] Benjamin Roth, Grzegorz Chrupała, Michael Wiegand, Mittul Singh, and Dietrich Klakow. Generalizing from freebase and patterns using cluster-based distant supervision for kbp slot-filling. 2012.

[80] Benjamin Roth, Tassilo Barth, Grzegorz Chrupała, Martin Gropp, and Dietrich Klakow. Relationfactory: A fast, modular and effective system for knowledge base population. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 89–92, 2014.

[81] Benjamin Roth, Tassilo Barth, Michael Wiegand, Mittul Singh, and Dietrich Klakow. Effective slot filling based on shallow distant supervision methods. *CoRR*, abs/1401.1158, 2014. URL `http://arxiv.org/abs/1401.1158`.

[82] Dan Roth and Wen-tau Yih. Global inference for entity and relation identification via a linear programming formulation. *Introduction to statistical relational learning*, pages 553–580, 2007.

[83] Arthur L Samuel. Some studies in machine learning using the game of checkers. iirecent progress. In *Computer Games I*, pages 366–400. Springer, 1988.

[84] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[85] Michael Schmitz, Robert Bart, Stephen Soderland, Oren Etzioni, et al. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534. Association for Computational Linguistics, 2012.

[86] Wei Shen, Jianyong Wang, and Jiawei Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460, 2015.

[87] Stephen Soderland, Natalie Hawkins, John Gilmer, and Daniel S Weld. Combining open ie and distant supervision for kbp slot filling. In *TAC*, 2015.

[88] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.

[89] Mihai Surdeanu and Heng Ji. Overview of the english slot filling track at the tac2014 knowledge base population evaluation. In *Proc. Text Analysis Conference (TAC2014)*, 2014.

[90] Mihai Surdeanu, David McClosky, Julie Tibshirani, John Bauer, Angel X Chang, Valentin I Spitkovsky, and Christopher D Manning. A simple distant supervision approach for the tac-kbp slot filling task. In *TAC*, 2010.

[91] Mihai Surdeanu, Sonal Gupta, John Bauer, David McClosky, Angel X Chang, Valentin I Spitkovsky, and Christopher D Manning. Stanford's distantly-supervised slot-filling system. In *TAC*, 2011.

[92] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 455–465. Association for Computational Linguistics, 2012.

[93] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077. International World Wide Web Conferences Steering Committee, 2015.

[94] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.

[95] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, Sandro Athaide Coelho, Sören Auer, and Andreas Both. Agdistis-graph-based disambiguation of named entities using linked data. In *International semantic web conference*, pages 457–471. Springer, 2014.

[96] Kenton Varda. Protocol buffers: Googles data interchange format. *Google Open Source Blog, Available at least as early as Jul*, 72, 2008.

[97] Ashish Vaswani, Yonatan Bisk, Kenji Sagae, and Ryan Musa. Supertagging with lstms. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 232–237, 2016.

[98] Guanying Wang, Wen Zhang, Ruoxu Wang, Yalin Zhou, Xi Chen, Wei Zhang, Hai Zhu, and Huajun Chen. Label-free distant supervision for relation extraction via knowledge graph embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2246–2255, 2018.

[99] Shaolei Wang, Yue Zhang, Wanxiang Che, and Ting Liu. Joint extraction of entities and relations based on a novel graph scheme. In *IJCAI*, pages 4461–4467, 2018.

[100] Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. Classifying relations via long short term memory networks along shortest dependency paths. In *proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1785–1794, 2015.

[101] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. Joint learning of the embedding of words and entities for named entity disambiguation. *arXiv preprint arXiv:1601.01343*, 2016.

[102] Xuchen Yao and Benjamin Van Durme. Information extraction over structured data: Question answering with freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 956–966, 2014.

[103] Dian Yu and Heng Ji. Unsupervised person slot filling based on graph mining. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 44–53, 2016.

[104] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, 2014.

[105] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762, 2015.

[106] Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 506–514, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P18-1047`.

[107] Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 506–514, 2018.

[108] Dongxu Zhang and Dong Wang. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*, 2015.

[109] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, 2017.

[110] Suncong Zheng, Yuexing Hao, Dongyuan Lu, Hongyun Bao, Jiaming Xu, Hongwei Hao, and Bo Xu. Joint entity and relation extraction based on a hybrid neural network. *Neurocomputing*, 257:59 – 66, 2017. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2016.12.075. URL `http://www.sciencedirect.com/science/article/pii/S0925231217301613`. Machine Learning and Signal Processing for Big Multimedia Analysis.

[111] Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. Joint extraction of entities and relations based on a novel tagging scheme. *CoRR*, abs/1706.05075, 2017. URL `http://arxiv.org/abs/1706.05075`.

[112] Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. Joint extraction of entities and relations based on a novel tagging scheme. *arXiv preprint arXiv:1706.05075*, 2017.

[113] Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. Doser-a knowledge-base-agnostic framework for entity disambiguation using semantic embeddings. In *European Semantic Web Conference*, pages 182–198. Springer, 2016.