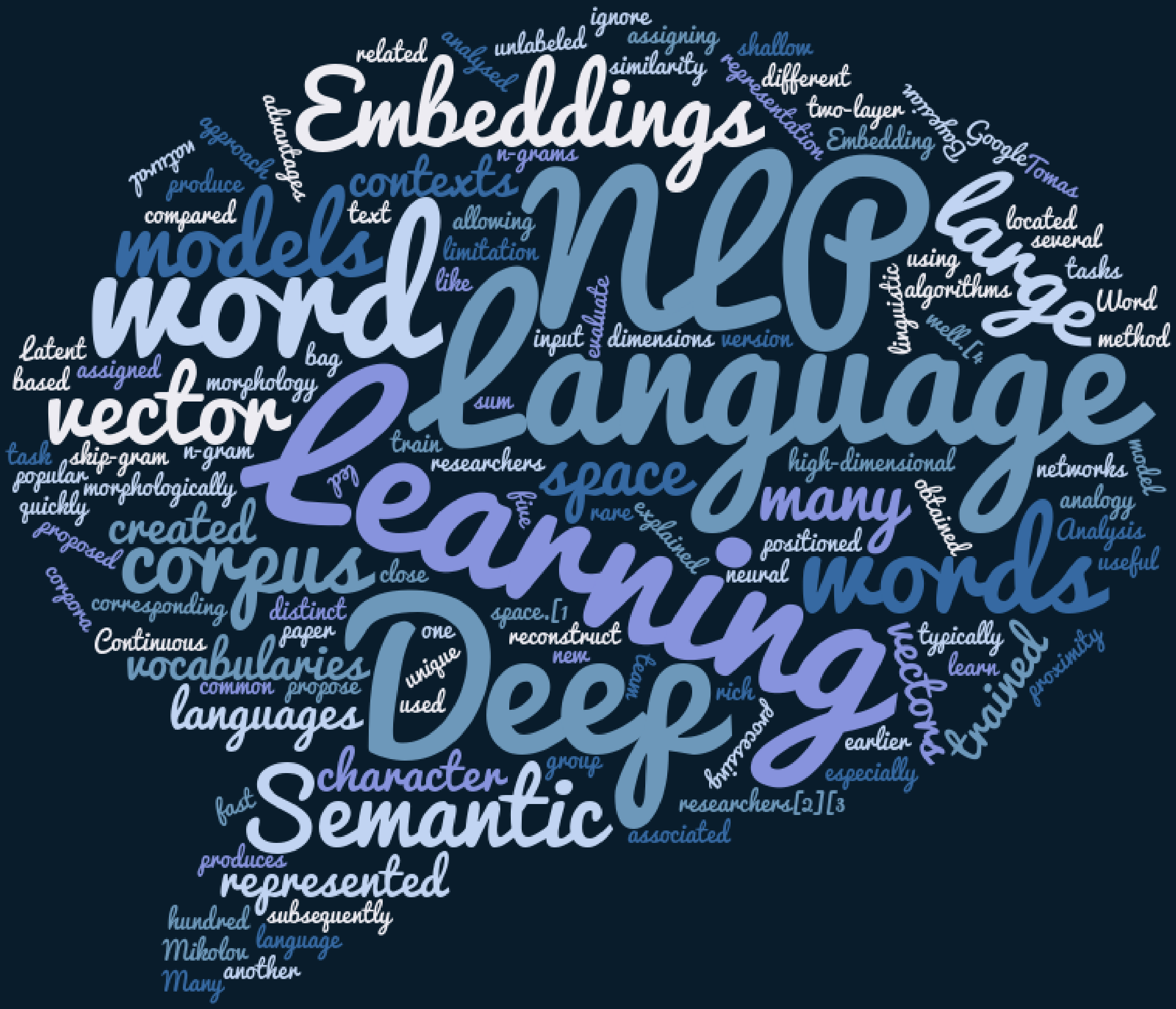
Language-consistent Open Relation Extraction

from Multilingual Text Corpora

T. Harting

Technische Universiteit Delft



TU Delft

Language-consistent Open Relation Extraction

from Multilingual Text Corpora

by

T. Harting

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday July 12, 2019 at 09:30 AM.

Project duration:	November 12, 2018 – July 12, 2019	
Thesis committee:	Prof. dr. ir. G. J. P. M. Houben,	TU Delft, chair
	Dr. rer.nat. C. Lofi,	TU Delft, supervisor
	Dr. ir. W.P. Brinkman,	TU Delft

An electronic version of this thesis is available at http://repository.tudelft.nl.
Cover image from http://www.deep-solutions.net/blog/wordembeddings.

TU Delft

Preface

Before you lies the thesis *Language-consistent Open Relation Extraction from Multilingual Text Corpora*. This research is aimed at extracting relations between entities from texts of many different languages. The report you are currently reading describes the work that has been conducted over multiple months, between November 2018 and June 2019. It has been written to obtain the degree of Master of Science at the Delft University of Technology, within the Computer Science programme.

When I first proposed my potential research topic to my supervisor, Dr. Christoph Lofi, he told me that this particular research domain could be seen as “the Wild West of Computer Science”. Many research efforts have been conducted, but the field is still maturing and finding a clear line forward. I instantly liked the analogy, since it meant that there were a lot of potential improvements that could be made. After a seemingly long process of diving into existing literature, figuring out other people’s code, writing my own code, fixing bugs, training and testing language models and wishing for good results, I am happy with, and proud of, the final result.

I would like to thank my supervisor for his guidance, time and valuable feedback on my research over the past months. I enjoyed the experience and always left our meetings with a head full of ideas. Of course, I also thank all others who took the time to guide me through the research process and provided me with helpful comments. A special note of gratitude goes out to my parents, who have been nothing but supportive during my time in Delft and who have enabled me to study there in the first place. Last but surely not least, I thank my girlfriend and friends for all moments outside of the lecture halls. They have made my time as a student truly unforgettable.

I am glad that you are taking the time to read my thesis and I wish you a pleasant reading.

T. Harting
Delft, June 2019

Abstract

Open Relation Extraction (ORE) aims to find arbitrary relation tuples between entities in unstructured texts. Even though recent research efforts yield state-of-the-art results for the ORE task by utilizing neural network based models, these works are solely focused on the English language. Methods were proposed to tackle the ORE task for multiple languages, yet these works fail to exploit relation patterns that are consistent over languages. Moreover, they require additional data to train translators, hindering efficient extension to new languages. In this work, we introduce a Language-consistent Open Relation Extraction Model (LOREM). By adding a language-consistent component to the current state-of-the-art open relation extraction model, we enable exploitation of information from multiple languages. Since we remove all dependencies on language-specific knowledge and external NLP tools such as translators, it is relatively easy to extend our model to new languages. An extensive evaluation performed on 5 languages shows that LOREM outperforms state-of-the-art monolingual and cross-lingual open relation extractors. Moreover, experiments on low- and even no-resource languages indicate that LOREM generalizes to other languages than the languages that it is trained on.

Contents

1

Introduction

In recent decades, the amount of openly accessible, human-written text has increased enormously. This rapid expansion is mainly caused by the explosive growth of the internet. Analysts approximate that the amount of digitally stored data overtook the amount of analogue data in 2002 and 94% of our data is stored in digital format since 2007 [37]. Due to the sheer volume of the available texts, it has become an infeasible task to manually process all this data. Therefore, a substantial body of research investigates ways to automatically process this textual data, in order to extract valuable information. This research field came to be known as Natural Language Processing (NLP). An important sub-task within the field of NLP is Information Extraction (IE). Within IE, we seek to extract structured information from *unstructured* or *semi-structured* texts. Unstructured text does not contain an identifiable structure and does not fit in relational tables [11]. Semi-structured texts also do not conform to a formal structure associated with a relational table of other forms of data tables, but nonetheless contain other markers and tags [70]. Examples are web pages, e-mails and PDF documents. The extracted structured information which forms the output of IE-systems can be used in a wide variety of applications.

Information Extraction itself is comprised of multiple sub-tasks. In this thesis, we will focus on one of these sub-tasks, i.e. Relation Extraction (RE). Culotta et al. [22] define RE as the process of discovering semantic connections between entities in unstructured texts. Looking at the literature in this domain, we find that there exists an active community which proposed a vast array of different methods for dealing with the RE-problem independent of the text domain. Two main paradigms of RE are identified; *closed* and *open* relation extraction. They differ in the sense that we possess a pre-defined set of relations that we are looking for in the closed paradigm, while the relations that we are looking for are unknown in the open setting. Given the highly heterogeneous nature of internet data, we are mainly interested in the open paradigm. Nonetheless, various interesting techniques were presented in the closed setting that could prove to be useful when applied to the task of open RE. Therefore, both paradigms will be discussed in this thesis.

From the literature, we can conclude that the vast majority of relation extractors is created for English texts. Although being a rough estimate, various cross-over studies imply that around 70% of the internet is written in languages other than English [77]. This suggests that more general methods that can easily be extended to the range of languages found in internet texts are needed. For this purpose, several multilingual relation extractors were proposed. Yet the scope of multilingual works juxtaposes the large body of English-based works and accordingly, improvements can be made. Additionally, current literature indicates that utilizing relation information that is consistent over languages can lead to performance gains. Given these observations, we define the research topic of this Master's thesis as *Language-consistent Open Relation Extraction from Multilingual Text Corpora*. Our main contributions are as follows:

- We provide an extensive literature overview on the field of relation extraction in the closed, open and multilingual domain.
- We introduce a Language-consistent Open Relation Extraction Model (LOREM). LOREM is the first open relation extractor that utilizes language-consistent relation structures to improve open relation extraction performance across multiple languages. In addition LOREM does not depend

on language-specific knowledge or external NLP tools such as translators or dependency parsers, thus allowing for easy extension to new languages.

- We are the first to employ multilingual, aligned word embeddings as the input of a multilingual relation extractor. Our experiments show that this improves the performance over using conventional monolingual word embeddings.
- We present experimental results on five high-resource languages showing that LOREM outperforms multiple state-of-the-art mono-lingual and cross-lingual open relation extractors. Additionally we present experiments on no- and low-resource languages which indicate the ease and effectiveness of expanding LOREM to additional languages.

This thesis is structured as follows. In Chapter 2, we describe the problem of relation extraction and its applications. Subsequently, we take a more detailed look into closed relation extraction in Chapter 3. Since we are primarily interested in open relation extraction, we will only briefly touch upon the history of the closed paradigm. Our attention will be mostly directed to state-of-the-art approaches and novel ideas. These could prove to be helpful when applied in the open setting. After the closed paradigm is discussed, we will continue with the open paradigm in Chapter 4. The paradigm is further introduced, after which state-of-the-art methods are described. Post-processing steps and evaluation methods will also be discussed. The final part of our literature review, multilingual relation extraction, will be discussed in Chapter 5. Two multilingual techniques, and their state-of-the-art instantiations, will be considered. Moreover, we describe multilingual datasets and multilingual word embeddings. From the literature review, we determine problems that remain to be tackled in this research field. In Chapter 6, we describe our model that provides solutions to some of these challenges. Chapter 7 is dedicated to validating our main hypothesis which entail the expected behaviour of our model, using a range of experiments. In Chapter 8, we provide a summarizing conclusion and discussion on our work. We complete this thesis by presenting our view on interesting future research on the matter in Chapter 9.

2 Problem Description

In this chapter, we first describe the more general problem of Information Extraction. From there, we zoom in on the relation extraction task, discussing its definitions and paradigms. We then define the main research questions of this work and finish this chapter by describing evaluation metrics and example applications.

2.1. Information extraction

Information Extraction (IE) is defined as the process of automatically extracting structured information from text [43]. The extracted structured information which comprises the output of IE-systems can be used in a wide variety of applications, of which we will mention some examples at the end of this chapter.

Although it is far from solved, the task of Information Extraction is not a new one. It has a history that dates back to the late 1970's [19]. Despite these early research efforts, the structured development of IE started in the late 1980's. At that time, the scientific interest in IE-systems grew and research projects were invoked. These projects were discussed at conferences, like the Message Understanding Conference (MUC)[1], which led to further developments of the field. Not only the conferences themselves proved to be helpful, the evaluation competitions that were proposed by the organizers of these conferences also had a substantial impact. These competitions allowed researchers to develop a solution for a clearly defined task, using data that was prepared by the conference organization. These tasks and data also came with an evaluation framework in which the proposed solutions could be compared against works of other research groups, adding a competitive aspect to the IE task. After 1998, MUC was succeeded by the NIST Automatic Content Extraction (ACE) conference[2].

Traditionally, the IE task is depicted as a pipeline system, in which multiple pieces of information are extracted in consecutive steps. In their work, Jurafsky et al. [43] give an overview of the typical steps that occur in such a system. The IE systems start by identifying named entities in the unstructured texts. These named entities can be persons, locations, companies, events, etc. The task of identifying and tagging these named entities came to be known as Named Entity Recognition (NER). As an illustrative example, a NER system might take in the following input sentence

Marcus went to a Pearl Jam concert in the Ziggo Dome.

and derive the following output

$[\textit{Marcus}]_{Person}$ *went to a* $[\textit{Pearl Jam concert}]_{Event}$ *in the* $[\textit{Ziggo Dome}]_{Location}$.

The field of NER is, and has been, a very active research domain in which a wide variety of techniques were proposed. In a co-reference resolution step, different linguistic realisations of the same

[1] https://www-nlpir.nist.gov/related_projects/muc

[2] https://www.ldc.upenn.edu/collaborations/past-projects/ace

named entities that were tagged in the first step are identified. This step can be seen as a post-processing effort which is crucial for yielding more accurate results [46]. Although being a very interesting sub-task, NER and its post-processing fall outside the scope of this Master's thesis. The third IE step in the proposed pipeline is Relation Extraction (RE), which is the main subject of this Master's thesis. Within RE, we seek to extract semantic relations between the tagged named entities in the given corpus. The fourth and final step in the IE pipeline is a post-processing step in which the extracted information is analysed in further detail. Even though these post-processing methods will be succinctly described in this literature review, their application will be a topic for future work.

Despite the fact that this traditional pipeline model is still widely used within IE literature, current works show that deviating from this model by combining different steps can have numerous advantages. That being said, this pipeline model does provide an intuitive perspective on the origin of the RE task. Moreover, it provides a baseline for current state-of-the-art methods to expand upon, which will also be discussed later on in this literature review.

2.2. Relation extraction

As stated, Relation Extraction can be defined as the process of discovering semantic connections between entities in unstructured texts [22]. Looking at the extensive body of literature within this research domain, we can identify different paradigms.

The first major distinction that we find in these works is the difference between binary and *n*-ary relations. The predominant part of current RE research is focussed on binary relation extractors. In these scenarios we can formally describe the RE task as the search for a set of tuples <Entity$_1$, Relation, Entity$_2$>, given an unstructured text in which the entities are tagged and where Relation describes a semantic connection between Entity$_1$ and Entity$_2$. We call Entity$_1$ and Entity$_2$ the relation entities in the input sentence. As an example, we would find the tuple <Alan Turing, deciphered, Enigma machine> from the sentence *"During world war II, Alan Turing deciphered the Enigma machine which was used to send secret messages."*.

Focussing on *n*-ary relations is a more general approach, were we describe a relation between entity sets, instead of between two entities. Formally, this can be described as looking for a set of tuples <{EntitySet$_1$}, Relation, {EntitySet$_2$}>, again given an unstructured text in which entities are tagged and where Relation describes a semantic connection between the entities in EntitySet$_1$ and EntitySet$_2$. Altering our example, we would find a tuple <{Alan Turing, Berkeley team}, deciphered, {Enigma machine, other German codes}> from the sentence *"During world war II, Alan Turing and the Berkeley team deciphered the Enigma machine and other German codes."*. In this thesis, we limit ourselves to the binary approach, which could be expanded in future work.

Within the binary approach, we can again identify two paradigms from the published literature; the *closed* and the *open* paradigm. In the closed paradigm, the set of possible relations that we are looking for is defined up front. Therefore, models are constructed which focus on identifying these specific relations (example relations are BornIn or GraduatedFrom). In contrast, the relations that need to be extracted are not pre-defined in the open paradigm. Naturally, solving this problem requires a different set of solutions. Typically, the relationship between two entities is directly extracted as as substring of the input sentence. Where a closed approach would extract <Turing, BornIn, England> from the input sentence *"Turing was born in England"*, an open approach would extract <Turing, was born in, England>. We will give an in-depth description and discussion on both paradigms in the next chapters in order to substantiate our choice for using the open paradigm in the rest of this thesis.

The vast majority of both open and closed relation extractors are created for the English language. The features and patterns that are used by these systems to find relations are specific to English language structures. As was advocated in the introduction, multilingual relation extractors are needed to cope with the wide range of languages that can be found on the web. We define multilingual relation extraction as the collection of two techniques; cross-lingual relation extraction and language-consistent relation extraction. Both techniques will be discussed in Chapter 5. In short, cross-lingual systems use information and tools from a target language to extract relations from a source language. Language-consistent systems, harvest information from multiple languages to create a model which can be applied to a range of languages. We need to take into account that not only high-resource languages, but also low-resource languages are used on the web.

2.3. Research questions

Given the the heterogeneous nature of internet data in both language and text domains, we observe a need for relation extractors which are not bound to a limited set of pre-defined relations, nor by one single language. Accordingly, the main problem that will be discussed in this work is that of multilingual open relation extraction. Yet, the number of works presented in the closed RE paradigm juxtaposes that of the open and multilingual paradigms. Moreover, the open and multilingual paradigms find their origin in the closed paradigm. Therefore, this paradigm will also be extensively discussed in this thesis. We define the main research questions that will be discussed in this thesis as follows.

RQ1: What is the state-of-the-art of *closed* relation extraction?

RQ2: What is the state-of-the-art of *open* relation extraction?

RQ3: What is the state-of-the-art of *multilingual* relation extraction?

RQ4: How can we combine state-of-the-art open and multilingual relation extraction research to obtain a *multilingual open* relation extraction model?

RQ5: How does the considered *multilingual open* relation extraction model hold on real-world test data?

To answer these questions, we conduct an extensive literature review on traditional and state-of-the-art relation extraction literature in the following chapters. Using our literature review, we will propose improvements that can be made to multilingual open relation extraction. These proposals will likely include possible combinations of currently existing deep learning methodologies that were proposed in this field. Widely used community datasets in multiple languages are used to verify the performance of the considered systems. To mimic real-world scenarios, we will evaluate our work on high-, low- and no-resource languages, which will be clearly defined later on. Common evaluation metrics are used to compare our work to the existing literature.

2.4. Evaluation metrics

Multiple benchmark datasets are used for evaluating the performance of closed, open and multilingual RE systems. Since these paradigms use different sets of benchmark datasets, we will discuss the specifics of the sets in the closed, open and multilingual RE chapters of this thesis, respectively. In contrast to the benchmark datasets, the metrics that are used during evaluation are the same in almost all published works. In the interest of comparability, we use this set of metrics to evaluate the added value of our research. These evaluation metrics are derived from two basic metrics; the *precision* and the *recall*.

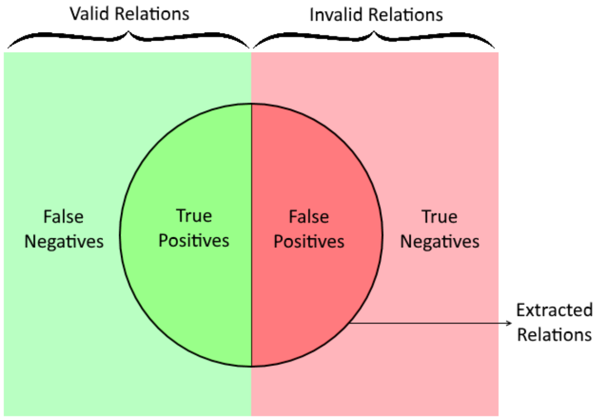


Figure 2.1: Illustration of True/False Positves (TP/FP) and True/False Negatives (TN/FN) in RE.

The precision is defined as the percentage of valid relations from the total extracted relations. Using Figure 2.1, we can formally define the precision as

$$P = \frac{TP}{TP + FP}. \tag{2.1}$$

The recall of a model is the percentage of extracted valid relations from all valid relations that are present in the input text. Using Figure 2.1, we can formally define the recall as

$$R = \frac{TP}{TP + FN}. \tag{2.2}$$

A frequently used metric that is derived from the precision and the recall is the F_1-score. The F_1-score is defined as the harmonic mean of the precision and the recall. It is used as a measure for the total performance of a relation extractor. Mathematically,

$$F_1 = \frac{2 \cdot P \cdot R}{P + R}. \tag{2.3}$$

2.5. Applications

Relation extractors are used in a diverse range of applications. We succinctly describe some example applications, to gain an insight on how these systems are used. As a first example, relation extraction is used to construct and complete knowledge bases [5]. Constructing these knowledge bases is a labour-intensive task that requires time and domain knowledge. At the same time, web pages contain extensive information that could be useful for this task. Using relation extraction and named entity recognition, one can extract this information from web pages in such a form that it can be added to the knowledge base. Another interesting application is question answering (QA). Relation extraction forms a key component of many QA systems (e.g. Yao et al. [89]). For example, in processing the question*"What is the birthplace of Alan Turing?"*, a QA system might search for a tuple `<Alan Turing, BornIn, ?>`. Relation extraction is extensively used in the medical domain. For instance, Chun et al. [85] use relation extraction to find gene-disease relationships in medical texts. Many other clinical applications exist [82]. As a last example, enterprises also benefit from relation extraction models. Laclavik et al. [47] propose a relation extraction model which provides a range of insights from enterprise emails, a text source that potentially holds valuable information. These examples show that there exist a diverse set of use cases for relation extraction, underlining the importance of well-performing models.

3

Closed Relation Extraction

In this chapter, we discuss research efforts that have been made within the closed RE paradigm, for the purpose of answering research question 1; What is the state-of-the-art of closed relation extraction? First, we give a concise historical overview. We then turn our attention to current state-of-the-art closed RE models. Next, used datasets and important weakly-supervision paradigms called bootstrapping and distant supervision are described. Last, we derive a conclusion on the usability of these works for the rest of this thesis, since our primary focus is on multilingual open RE.

3.1. From hand-crafted rules to machine learning

Initially, research efforts focused on applying manually crafted linguistic rules and patterns to extract relationships from text [69]. These traditional approaches have two major drawbacks [46]. First and foremost, manually creating these extraction rules is a labour-intensive task that requires extensive work by linguists and domain experts. Given the enormous amounts of text that is currently available on the web, the traditional approach does not provide a scalable, and therefore viable, solution. The second drawback is that most rule-based methods are highly dependent on the text domain in which they were created and even more so on the language in which these texts are written. Since texts from the internet contain a highly diverse set of domains, it can again be concluded that traditional methods do not scale to current day applications. There exist some domain-independent rule-based works (e.g. Hearst [35]). However, these approaches typically only apply to a set of relations (only hyponyms in Hearst's work [35]). They do not work for all relation types. Konstantinova [46] concludes that these traditional methods are only a valid option when the main aim is to quickly get results from well-defined domains and text corpora.

3.1.1. Multi-class classification

Given these serious limitations, research efforts focussed increasingly more on statistical machine learning approaches in the early 2000's. These methods automatically learn extraction patterns from labelled datasets. This has the obvious advantage that it scales to large corpora of texts. Furthermore, machine learning-based efforts were made to create well-performing domain-independent models, which will be described later on in this chapter. The closed RE task is typically approached as a multi-class classification problem. Here, the pre-defined set of relations are the classes that can be predicted from an input sentence in which two entities are tagged. In many approaches, an `other` class is added to the set of possible relations to allow the model to classify relations that are not part of the given relation set.

In the literature, numerous different machine learning models have been used to solve this multi-class classification task. Examples include kernel methods [21, 92], maximum entropy models [44], support vector machines [34] and conditional random fields [22]. Given the different characteristics and strengths of these statistical models, they yield different evaluation results.

3.1.2. Lexical and syntactic features

An arguably more interesting aspect of these research works is their choice of model input, also called features. Machine learning models almost exclusively deal with numerical input data. Since the relation extraction task is defined on natural text, providing valid input for the models is a non-trivial task. Accordingly, efforts were made to extract features that can be easily encoded in a numerical format (e.g. categorical features). We can identify two types of features that are used, being lexical features and syntactic features.

Lexical features can be extracted from the input text with very high efficiency. Therefore, some works refer to them as shallow or basic features. Instances of these lexical features are

- the length of the input sentence,
- the number of words separating the two entities,
- the mention level of both entities (`Name`, `Pronoun` or `Nominal`)
- and overlap features (e.g. a flag indicating which entity came first in the sentence).

In addition to lexical features, syntactic features are explored. In order to extract these features, the sentences from the used corpus are parsed, resulting in a so called parse tree. As an example, Figure 3.1 depicts the parse tree extracted from the sentence *"the artist stood on the stage"*. In this figure we use the symbol *S* as the root symbol, *N* for nouns, *V* for verbs and *P* for preposition words. From these parse trees, features such as the PoS-tags and entity types of the words on which the relation entities are dependent can be used in the hope of training better extractors. As is often the case, this added complexity comes at the cost of a significant efficiency loss because of the increased computational complexity. Using only lexical features can result in a model that runs 30 times faster than using syntactic features [86].

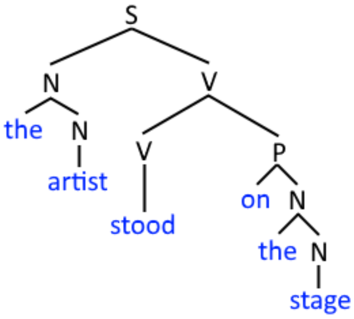


Figure 3.1: Example parse tree for *"the artist stood on the stage"*.

In two extensive surveys [34, 40], the performance of lexical and syntactic features is empirically investigated in order to find out if the improvements gained by adding syntactic features are worth the efficiency loss. Both surveys derive the same conclusion; adding syntactic features only marginally improves the performance. Consequently, this limited advantage does not weigh up to the loss in efficiency. That being said, it is suggested in both works that this limited effect is (partly) caused by the dataset that is used. Both surveys use the ACE dataset[1] for evaluation purposes. The writers find that most relations defined in this dataset have two entities that lay close to each other in the sentences. These so called short-distance relations can most likely be resolved using shallow features. Hence no significant improvements can be gained by adding syntactic features. Other works indeed imply that the parse-based features can be highly informative when longer and more complex sentences are present in the corpus [86]. Concluding, the literature indicates that shallow lexical features should be used when only relatively simple sentences are present in the corpus. When more complex and longer sentences appear, adding syntactic features can yield significant performance gains at the cost of efficiency.

[1] https://www.ldc.upenn.edu/collaborations/past-projects/ace

3.2. From machine learning to deep learning

As we have seen, hand-crafted feature development for relation extraction relies heavily on existing NLP techniques, such as PoS-tagging, named entity recognition and dependency parsing. Using these tools allows the RE model to inherit the discovered knowledge from these techniques. Nonetheless, Bach et al. [6] advocate that this dependency hinders the performance of RE systems since errors made by the NLP tools accumulate downstream. Moreover, these tools might not capture all the necessary information for the RE task and the proper NLP tools might not be available for all languages of interest.

3.2.1. Deep neural networks

The problem of error propagation from pre-processing steps is not limited to the RE task. In fact, Collobert et al. [18] state that this problem occurs in a wide range of NLP models. For this reason, they propose a pivot from using existing NLP tools to Deep Neural Network (DNN) based approaches. Given enough data is available, these DNN approaches can automatically learn the relevant features, bypassing the error-prone feature engineering step.

These DNNs for relation extraction broadly entail three steps, which are depicted in Figure 3.2. First, we still need to find a numerical representation for the text that we use as input for the system. This should however be done without relying on external NLP tools trained on labelled datasets. For this purpose, an ingenious technique named word embeddings is used. This technique will be specified in Section 3.2.2. Once this step is completed, a DNN architecture is trained to extract sentence-level features from the input. From the literature, we see that two DNN architectures are commonly used for this purpose; Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). Both will be discussed in Section 3.2.3. Finally, the sentence-level representations are used to classify the relations from a labelled training dataset.

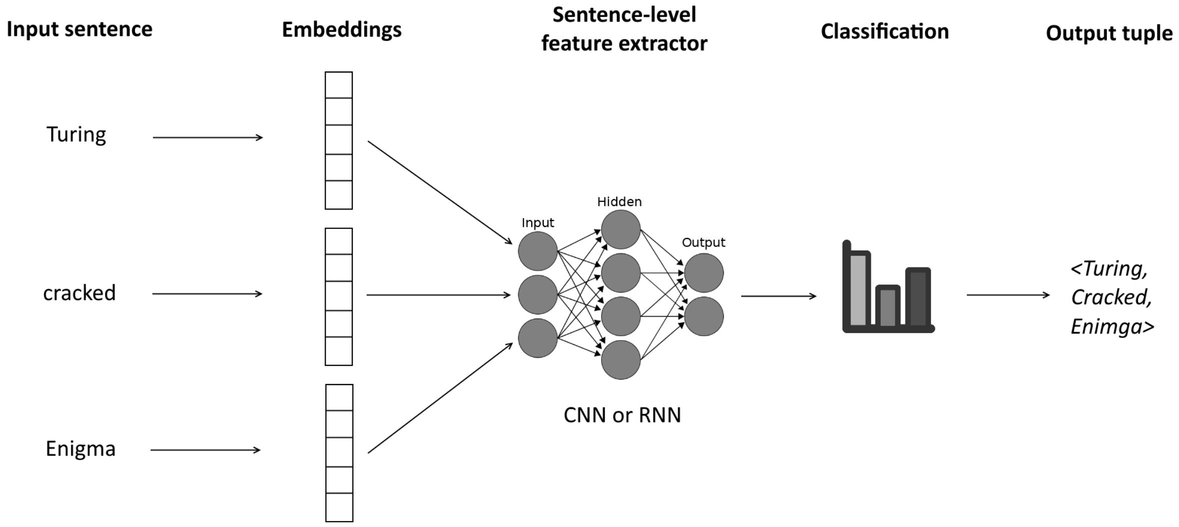


Figure 3.2: Deep neural closed relation extraction framework.

Notwithstanding the aforementioned benefits, the introduction of DNNs for the relation extraction task also has an obvious disadvantage that is rarely discussed in neural RE literature. Given the complexity of these DNNs, they are often regarded as black box algorithms. This means that we do not know why the model makes a certain decision, we only see its output. Even though recent works have shown that it is possible to extract meaning from intermediate layers of DNNs (e.g. Zeiler et al. [91]), comparable studies have not been conducted within the RE domain. This can be explained by the fact that the need for understanding the model's decision is lower than in many other domains. For example, understanding the reasoning of the model can be vital for medical systems. Yet, knowing why the model extracts a certain relation is typically not vital knowledge. That being said, gaining insights into the reasoning of the model could prove to be helpful for identifying possible improvements. Therefore, we suggest this could be an important direction for future work.

3.2.2. Embeddings

Just like the discussed machine learning models, DNNs also require numerical input data. The input needs to be represented in such a way that the DNN can implicitly extract features from it, without relying on external tools that are trained on labelled and domain-specific data. Even without using these NLP tools, we want to capture the semantic and syntactic information from the text in a numerical representation. A range of embeddings was proposed exactly for this purpose. The vast majority of state-of-the-art neural relation extractors use one or more of these embeddings as their input. It is important to note that the use of these embeddings is not limited to DNNs, they can also be used in alternative machine learning models.

Word embeddings

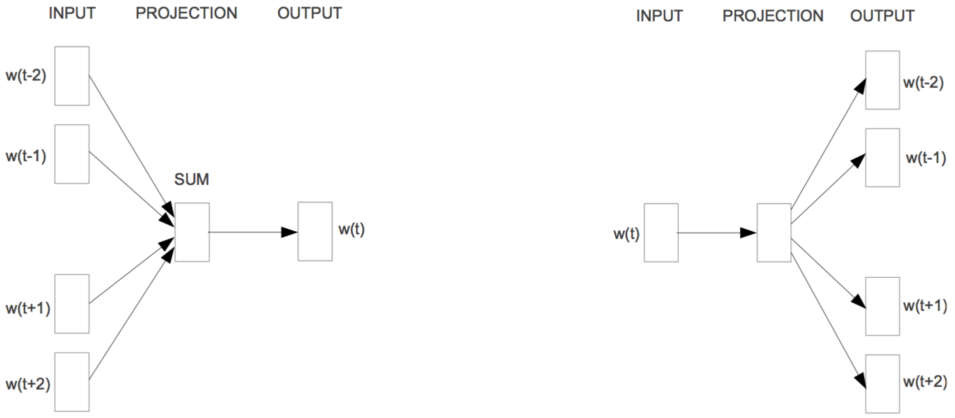
The most frequently used embedding type within NLP is the word embedding, in which each separate word is represented as a dense vector. The most eminent idea behind these word embeddings dates back as early as 1957. In that year, J.R. Firth proposed that a word is known by the company it keeps [29]. This means that we can describe a word by other words that it is frequently surrounded by. Consequently, two words that often appear in the same context are expected to have similar semantic meaning.

Initially, Brown clusters were used to incorporate this idea into word representations [14]. Here words that appeared in a similar context would be placed in the same clusters, which were then used to form the word vectors. Although its simplicity makes this an attractive approach, the usage of Brown clusters is limited by its computational complexity. Training such a model on large corpora is infeasible, since the complexity scales quadratically with the number of words in the corpus [63]. Multiple works that present more scalable word representations were published, most notably the introduction of neural network-based approaches [10]. Following the concept of J.R. Firth, a feed-forward neural network is trained to predict a word given *n* previous words. No labelled data is needed to train this model, one can simply use any available text corpus and sequentially try to predict each word given its *n* previous words. This model is still very expensive to train, mainly due to the non-linearity of the hidden layers. It did however form the basis for a breakthrough in word embeddings.

This breakthrough was achieved in 2013 by Mikolov et al. [57]. In this work, the writers present two simplified feed-forward neural network models. By replacing the non-linear hidden layer with a log-linear hidden layer, they drastically decrease the computational complexity of the model. Despite the fact that this alteration makes the model less effective in predicting the current word given *n* previous words, it does allow the model to be trained on much more data efficiently, resulting in better word embeddings.

The first model that is presented is the Continuous Bag of Words (CBOW) model. Besides the log-linear alteration, this model differs from the feed-forward neural network in another way. The CBOW model takes previous *and* next words into account to predict the current word, we call the surrounding words the context words of the current word. This results in the model depicted in Figure 3.3a. The second proposed model is the skip-gram model. This model is the exact inverse of the CBOW model. It tries to predict the context of a word, given that word. A visualization of this model can be found in Figure 3.3b. Extracting the word vectors from these models is straightforward once they are trained. We again refer to the idea that the semantic meaning of a word can be captured by the words that it is often surrounded by. If we project that idea on these models, we find that two words that have similar contexts should yield similar word embeddings. Concurrently, we know that the model is stimulated to give the same output of the hidden layer for two words that appear in the same context. From this, it can be concluded that the output of the hidden layer is actually the word embedding. The dimensionality of this word embedding (the number of neurons in the hidden layer) is a hyper-parameter that needs to be set in advance. Apart from CBOW and skip-gram, these models are also referred to as Word2Vec. Shortly after the original paper was published, Mikolov et al. presented several extensions that improve both the quality of the vectors and the training speed [58]. Despite these improvements, the general structure of these models remained the same.

Since its introduction, Word2Vec provided a fresh perspective on a wide range of NLP problems, yielding state-of-the-art-results. In practice, word embeddings are rarely trained from scratch. Nearly all RE models that use word embeddings initialize them with the ones pre-trained on a large, domain-



(a) The CBOW model.

(b) The skip-gram model.

Figure 3.3: The word embedding models as depicted by Mikolov et al. [57].

independent corpus. Many of these trained word embeddings are freely available. Even though it is clear that pre-trained embeddings are superior over training randomly initiated ones, there is a disagreement in the literature about the added value of updating these embeddings during training. Qu et al. [66] suggest that pre-trained word embeddings should not be updated while training the model for a specific task. They argue that this should not be done to avoid overfitting. On the other hand, Nguyen et al. [61] demonstrate that their model performs best when the word embeddings are allowed to vary during training to reach an effective state for relation extraction, providing evidence for non-static word embeddings. We can conclude that there exists no consensus on this topic in the literature.

Position embeddings

Zeng et al. [93] remind us that traditionally, structure features from parse trees were used in RE. They suggest that it is not possible to capture this structure information solely through word embeddings. In addition to word embeddings, the researchers propose an interestingly simple embedding called the position embedding. The concept behind this embedding is that the relation entities e_1 and e_2 need to be specified in the input sentence. For this purpose, the position embedding is constructed as a combination of the relative distances between the current word and e_1 and e_2. For example, the relative distances of *"on"* to *"artist"* (e_1) and *"stage"* (e_2) in the sentence *"the artist stood on the stage"* are 2 and -2 respectively. These relative distances are mapped to two vectors, for the distances to e_1 and e_2 respectively. The final position embedding of a word is defined as the concatenation of these two vectors. The dimensionality of these vectors is a hyper-parameter that needs to be set, as was the case for word embeddings.

Since position embeddings introduce extra variables in the model which have a negative impact on the efficiency, Wu et al. [86] consider a simpler alternative. Their so called indicator encoding is a **1** vector if the current word is a relation entity and a **0** vector otherwise. The experiments show that the position embeddings are crucial for CNN-based approaches, which is caused by the spatial invariance of CNNs according to Wu et al. In contrast, the researchers find that position embeddings only slightly improve indicator encodings for RNN-based models. This slight improvement does not justify the efficiency decrease, so they advice to use indicator encoding instead for RNN models. This is accounted to the capacity of exploiting temporal dependencies within RNNs. The same conclusion is reached by Dongxu Zhang et al [95]. They take a slightly different approach by placing markers (<e_1>, <e_2>, </e_1> or </e_2>) around the relation entities. The authors also conclude that the relative positional information can be implicitly obtained in the RNN, simply annotating the relation entities suffices. In short, position embeddings should be used for CNNs and entity markers suffice for RNNs.

Hand-crafted feature embeddings

Even though DNNs were initially proposed to bypass the problem of error propagation from NLP tools during feature engineering, several works reintroduce these tool-dependent features in the DNN setting. These works show that using these hand-crafted feature-based embeddings can improve the overall performance of the relation extractor on the used datasets [30, 49]. For example, Fu et al. [30] use entity type embeddings and on-depth-path embeddings, which rely on named entity taggers and tree parsers respectively. Naturally, the problem of error propagation is revived in these approaches and the risk of overfitting on the domain and language of the training data increases. Since we are interested in approaches that generalize well over heterogeneous internet data, word and position embeddings provide a better fit for our need.

3.2.3. DNN architectures

Now that we have discussed the text representation techniques, methods are needed to automatically extract informative sentence-level features from these representations and classify possible relations in the input texts. Towards that end, the existing literature proposes two main deep neural network architectures; convolutional neural networks and recurrent neural networks. Both models have different strengths and weaknesses, leading to different results.

CNN-based relation extractors

Zeng et al. [93] were the first to present a straightforward CNN-based relation extractor, it consists of three steps. First, words are represented by word and position embeddings. The word embeddings are initialized using a pre-trained word embedding set. Then, lexical- and sentence-level features are derived, which are concatenated to construct the final feature vector. There are five lexical-level features, being the word embeddings of both relation entities, their left and right neighbouring words and their possible WordNet hypernyms[2]. To create the sentence-level features, the word and position embeddings of all words in an input sentence are combined to form an input matrix. Since we want to find one relation class for each input sentence in which two entities are annotated, there is a need to merge the embedding matrix into a single sentence-level feature vector that can be used during classification. CNNs offer a natural method to facilitate this merging process. For this reason, Zeng et al. use convolutional layers to extract features from the embedding matrix. Additionally, max-pooling layers are used to determine the most relevant features and Rectified Linear Units (ReLU) are used to simplify the gradient calculation during training. Finally, the learned lexical- and sentence-level features are concatenated and fed into a softmax classifier to find the confidence scores for each relation. The whole process is trained using stochastic gradient descent-based back-propagation, which is the norm for many deep neural networks. From their results, Zeng et al. conclude that their system based on automatically learned features yields satisfactory results, it can therefore replace systems based on hand-crafted features that use external NLP tools.

In an improved version, Zeng et al. [94] adapt the original model in two ways. First, multi-instance distant supervision is applied. This technique will be discussed in Section 3.4.2. Second, the original max-pooling operation is adapted to provide a better fit to the relation extraction task. The initial CNN-model contained single max-pooling to capture the most significant features in each feature map that is generated by the convolutional layer. Zeng et al. show that, although single max-pooling is widely used, this approach is not a perfect fit for the RE task. According to the writers, single max pooling reduces the dimensionality too fast. Moreover, in relation extraction, a sentence is often split into three segments; words before the first relation entity, words between both entities and words after the second relation entity. Instinctively, a single max-pooling approach does not suffice to capture information from the separate segments. Therefore, Zeng et al. introduce the piecewise max-pooling operation, in which they return the maximum value of these three separate segments instead of one value per sentence. In their experiments, Zeng et al. demonstrate the significant improvements of their improved Piecewise Convolutional Neural Network (PCNN). The writers show that both alterations yield significant improvements, with the piecewise max-pooling yielding the biggest performance gain.

Notwithstanding the fact that the models presented by Zeng et al. [93, 94] massively decrease the reliance on external tools, external knowledge is still used in the form of knowledge bases. As we already discussed we are more interested in methods that are less dependent on external knowledge, since these naturally offer a more domain- and language-independent approach. In an effort to remove this external dependency, Nguyen et al. [61] propose a variation on the CNN-based relation extractor. The initial model is altered in three ways. After initialization, Nguyen et al. optimize both word and

[2] WordNet (`https://wordnet.princeton.edu`) is a large knowledge base that includes hypernyms for many different words.

position embeddings as model parameters. As discussed in Section 3.2.2, they advocate that this allows the embeddings to reach an effective state for relation extraction (at the risk of overfitting). Additionally, the word embeddings are initialized using a different set of pre-trained word embeddings. As a final alteration, Nguyen et al. allow the windows of the convolutional filters to vary in size, while Zeng et al. only use a single window size. This allows the model to capture a wider range of features which could be helpful for relation extraction. The presented experiments show that the performance of this model is highly similar to that of the initial CNN-based model by Zeng et al. Yet, this model does not depend on an external knowledge base. Nguyen et al. attribute this to the three alterations, but do not provide a more detailed analysis on the impact of the separate alterations. In conclusion, Nguyen et al. show that external tools are not a requirement for state-of-the-art relation extractors.

Santos et al. [25] reach an equivalent conclusion. They do however propose a disparate set of improvements. Similar to Nguyen et al., this model only uses word and position embeddings as the input, omitting any other external sources. It differs from the previously described models in the fact that the softmax classifier that is used to compute the confidence scores for each class is replaced by a ranking approach. In this approach, the CNN is used to create a vector representation for each class. Given an input sentence, a vector representation is created using the same CNN, which is then compared to each class vector representation. The main advantage of this technique is that the `other` class that is often introduced to capture relations that are not part of the given relation set can be easily omitted during training. Santos et al. suggest that this is a favourable characteristic, since the `other` class holds many different relations and is therefore very noisy. By simply omitting the vector representation of the `other` class, the training process focusses solely on the defined relation set. Experiments show that omitting this class during training has a significant positive impact on the F_1-score of all classes during testing, at the cost of a decreasing performance within the `other` class. Since we are typically more interested in the predefined set of relations within the closed paradigm, this is a valid trade-off to make.

Wang et al. [80] advocate that even more improvements can be made without relying on external NLP tools. They introduce two novel attention mechanisms for CNNs. The primary attention mechanism works directly on the input word and position embeddings. For both relation entities, a diagonal attention matrix is created to characterize the strength of the connection between the relation entity and each word in the sentence. The attention matrices are updated during the training of the network. From these matrices, we can compute the degree of relevance of each word to both entities. The final attention component for the input sentence is found by simply averaging the relevance degrees over both entities. This mechanism allows the model to automatically determine which parts of the sentence are most influential with respect to the two relation entities. The secondary attention mechanism applies a similar kind of attention matrix, which is also updated during training, on the max-pooling layer. Here, we want to select only the relevant parts of a sentence for the target relations, while neglecting the rest of the sentence. To summarize, the primary attention mechanism steers the network towards parts of the sentence that are most important to the specific relation entities and the secondary attention mechanism focusses on parts of the sentence that are most important for all target relations. The experiments show significant improvements are made by including these mechanisms. Moreover, there is room for more improvements. The last three models do not apply piecewise max-pooling, results imply that incorporating this technique would lead to better performing relation extractors.

RNN-based relation extractors

Initially, RNN-based relation extractors relied heavily on external knowledge and NLP tools like dependency parsers. Xu et al. [88] present a model which uses an external dependency parser to find the shortest dependency path between the two annotated relation entities. From this path, they extract four embeddings; word embeddings, PoS-tags, dependency types and WordNet hypernyms. Please note that these embeddings greatly depend on external knowledge. The four embeddings are used as the input for four RNN models, in order to automatically extract informative features. RNNs keep a hidden state vector which changes with the input at each step accordingly. This provides a natural solution to integrate long-range information within texts. It is however well known that a problem might arise when classical RNNs are trained, this problem is also encountered by Xu et al. Training a neural model requires gradient back-propagation. When the propagation path is too long, the gradient might grow or decay exponentially, leading to a vanishing or exploding gradient. To solve this problem, Long Short-Term Memory (LSTM) units were proposed [38]. This work introduces a gating mechanism that can be trained to automatically decide to what extend previous states and extracted features from the current step are memorized. Since their introduction, LSTMs became a vital part of most RNN models that are used in practice. An added max-pooling layer is used to gather information from the four RNNs and the resulting vectors are concatenated to form the input for the softmax classifier. The experiments show that this model performs slightly worse than the CNN-based model by Santos et al. [25]. Moreover, this model relies on a range of external tools and sources, where the CCN-based model only uses pre-trained word and position embeddings.

Miwa et al. [60] advocate a joint model of entity and relation extraction. They apply two bi-directional LSTMs, one for the embedded word sequence and one for the extracted dependency tree. A problem that occurs in one-directional LSTMs is that information from future words is not used when extracting features from words in the middle of a sentence. To solve this problem, one can create two LSTMs and provide the sentences in direct and inverse order as input. By simply adding up the results of both models, the final feature vector is obtained. Just like in Xu et al. [88], a range of external tools are used to create word, PoS, dependency type and entity label embeddings. The researchers make the somewhat surprising observation that, although entity label information is beneficial, the joint modelling of entity and relation extraction does not significantly improve either of these tasks. Nonetheless, using both word sequences and dependency trees is found to be effective and the F_1-score of Santos et al. [25] is slightly improved.

In later research, even more enhancements were proposed. Ammar et al. [3] extend Miwa's model by character-level embeddings and gazetteers. Whether this improves the performance of the model is not shown in their experiments. Additionally, Christopoulou et al. [17] propose a walk-based model which represents the input as a graph. This method removes the need for a dependency parser and includes the notion that multiple relations between different entities in a sentence can be dependent on each other. The presented F_1-scores are slightly worse than the scores yielded by Miwa et al. Although the dependency parser is no longer used, this model still depends on external knowledge in the form of named entity types.

Given the problems that arise when relying on external NLP tools and knowledge (domain- and language dependency, error propagation, etc.), independent models were proposed. Dongxu Zhang et al. [95] show that it is possible to create a competitive RNN model, using only word embeddings. In this model, words are encoded in word embeddings and entity markers are added as described in Section 3.2.2. Then a bi-directional RNN models the word sequence and produces word-level features, which are merged by a max-pooling layer to form the sentence-level feature vector. This may seem illogical, since RNNs can be used to create the sentence-level features in one step. However, the writers found that long-term information is lost quickly and the problem of gradient vanishing arises when using this approach. As we know from previous work, LSTMs could also provide a solution in these cases. Finally, the sentence-level features are used in a logisic regression softmax classifier. In their experiments, Dongxu Zhang et al. show that max-pooling, entity markers and bi-directional models all have a significant positive impact on the F_1-score, although they are still outperformed by CNN-based relation extractors.

We identified that LSTMs could provide a more suitable solution to the long-term information loss problem put forth by Dongxu Zhang et al. Shu Zhang et al. [98] show that this is indeed the case. These approaches are highly similar. The three main differences are that Shu Zhang et al. include LSTMs instead of the max-pooling layer, they only use word embeddings and no position embeddings or entity markers and they use a Multi-Layer Perceptron (MLP) instead of a logistic regression classifier. The presented results show that these alterations indeed improve the F_1-score on the same dataset, yet the writers do not analyse the impact of the separate differences. This model would likely benefit from placing markers around relation entities. Again, it is shown that adding PoS, named entity type, hypernym, position and dependency features to the model leads to performance improvements, with the risk of domain- and language dependency, error propagation, etc. When only word embeddings are used, the results of Shu Zhang et al. and Santos et al. [25] are highly similar.

CNNs versus RNNs

These works demonstrate that DNN-based relation extractors can yield satisfactory performance without relying on external knowledge and tools. In consequence, these models could provide a good fit for our heterogeneous, multilingual internet data. We find that state-of-the-art CNNs slightly outperform state-of-the-art RNNs on the used datasets. Nonetheless, the literature indicates different strengths of both neural networks, implying that there is no silver bullet for all RE tasks.

The main advantage of CNN-based relation extractors is that they typically have significantly less parameters that need to be learned than RNN-based methods [65]. Consequently, CNN-based approaches tend to be more efficient for the RE task. In the general NLP domain, RNNs have proven to be a good learner of word sequences. However, Qin et al. [64] argue that relation information is predominantly reflected in local features (e.g. trigger words) and not in global features (e.g. word sequences). Therefore, they advocate that CNNs are naturally more suitable for the RE task.

In contrast, Dongxu Zhang et al. [95] view this difference as a potential problem of CNN-based methods. They also state that CNNs can only learn local patterns, which means that learning long-distance relation patterns is not feasible using these models. This becomes a problem when we want to extract relations from a corpus that contains long sentences. One might argue that we could simply enlarge the window size of the convolutional layer to alleviate this problem. Dongxu Zhang et al. explain that this is not a valid solution, since the CNN will lose its strength of modelling short-distance patterns when the window sizes are too big. As was already discussed, Nguyen et al. [61] suggest that multiple window sizes could be used for this purpose. Even though this is a valid improvement, it has a significant impact on the efficiency of the model and tuning these window sizes is not a trivial task [95]. RNN-based models, especially those that contain LSTM units, offer an elegant solution for modelling these long-distance patterns. In general, natural text is interpreted as sequential data. In contrast to CNNs, RNNs are designed specifically for data that has a sequential nature. Notwithstanding the fact that CNNs can be altered to suit the RE task, RNNs seem to offer a more naturally fitting solution. Concluding, we find that CNNs offer a better solution when relation information is reflected in local features and RNNs form a better fit for the global feature scenario.

One might wonder if we could combine a CNN and RNN model to improve the RE system. Vu et al. [78] present multiple contributions to the RE task, of which combining both models is arguably the most innovative improvement. First, a CNN that uses max-pooling and varying window sizes is trained. To this end, Vu et al. use a novel representation by splitting the sentence in two parts; the words in front of the first relation entity, the relation entity and the words between both entities and the words after the second relation entity, the second relation entity and again the words between both entities. After these two contexts are processed by two independent CNNs, their resulting vectors are concatenated to form the final sentence-level representation. The writers argue that repeating the words between the two relation entities enforces the network to pay special attention to these words, assuming that they are more important than other words in the sentence. Second, a so called connectionist bi-directional RNN is trained. Next to a forward and backward pass, this network also trains a combination of both. An ensemble of CNN and RNN models are trained using only word and position embeddings and the ranking loss objective function presented by Santos et al. [25]. Finally, the models are combined using a simple voting process where the class with the most votes is picked. Since multiple models have to be trained, the computational complexity increases. The experiments show that the combined model outperforms all previously described models. This confirms the conclusion that CNNs and RNNs provide complementary information, therefore both approaches will be examined in this research work.

3.2.4. Additional improvements

In previous sections, we described a vast array of improvements on CNNs and RNNs for the closed RE task. Besides these model-specific improvements, the RE literature includes multiple additional improvements for DNN-based relation extractors.

Previously described methods tend to be biased towards the domain of the dataset on which they are trained (e.g. news corpus or sports reports). In order to adapt the relation extractor to another domain, one has to construct a labelled training set in that domain, which is a tedious task. To alleviate this problem, Fu et al. [30] propose a Domain Adversarial Neural Network (DANN). This model extracts sentence-level features using a DNN. In addition to the relation classifier, a domain classifier is added to the model. In contrast to the relation classification loss, the model is optimized to maximize the loss of the domain classifier. Since these classifiers are trained jointly, the network is encouraged to correctly classify the relations and incorrectly classify the domain at the same time. Hence, the sentence-level feature extractor should find features that are well suited to classify relations, while being ill suited to classify domains, resulting in domain independent features. The experiments indeed indicate an improvement in cross-domain relation extraction. This adversarial model could be an interesting approach to find not only domain independent, but also language independent features when a language classifier is added.

Another method to incorporate adversarial training is presented by Wu et al. [87]. In an effort to regularize (i.e. avoid overfitting) the relation classification algorithms, they add adversarial noise to the training data. This method was initially introduced by Goodfellow et al. [32], it is successfully applied to straightforward classification tasks like image classification. Wu et al. show that adding small adversarial noise to the word embeddings improves the robustness of relation extractors and leads to higher precision scores on different datasets.

Next to adversarial training, dropout strategies are also employed for regularization. Dropout was introduced by Srivastava et al. [73]. The key idea is that if we randomly drop units from the network during training, the network units do not co-adapt too much. For RNNs conventional dropout can not be applied directly, since this could harm the essential memorization characteristic of the LSTM units. Xu et al. [88] show that applying dropping word embeddings and units from the penultimate layer boosts the performance, while dropping inner LSTM cells is indeed inimical to the model.

Furthermore, Li et al. [49] demonstrate that unsupervised pre-training of the DNN model also has a positive impact on the performance. They introduce a sequence reconstruction loss function for this purpose. The DNN-based sentence-level feature extractor is used as a decoder. Once an input sentence is decoded, a LSTM-based decoder tries to reconstruct the original input sentence from the decoded representation. The loss function is defined as the discrepancy between the original and resulting sentence. After this pre-training step, the model is trained for relation extraction. Li et al. show that the pre-trained models yield similar performance to their counterparts that are not pre-trained, while using merely half or less of the training data.

The DNN model can also be pre-trained using a task that is similar to the task at hand, this technique is called transfer learning. Liu et al. [53] show that the performance of a neural relation extractor can by increased by first training the model on a named entity recognition task. The RE model is then initialized using the parameters from the NER model. Since these are related tasks, this initialization is more reasonable than a standard random initialization.

The described improvements can be implemented in both CNN- and RNN-based approaches. Given the generic nature of these techniques, they could also prove to be useful for DNN-based open relation extractors developed in this work.

3.3. Datasets

The discussed literature uses multiple freely available datasets for model training and testing. An overview of these datasets can be found in Table 3.1. Each of these datasets includes an `other` relation class to capture unknown relations. The vast majority of discussed works use either the SemEval-2010 Task 8 or ACE05 dataset. The main difference between these sets is that ACE05 is notably more biased towards the `other` class. 90.4% of its sentences are from this class, while only 17.4% of the SemEval-2010 sentences are from this class. Therefore, the SemEval-2010 Task 8 dataset is better suited for the RE task and is most frequently used for this reason. The ACE05 dataset does have the advantage that multiple languages are available, enabling multilingual training and testing. The StanfordRE dataset is specifically developed to include sentences that contain multiple relations, which means that a sentence can belong to multiple classes. The NYT dataset contains the least amount of sentences of the discussed datasets, while concurrently having the most classes. Accordingly, this dataset is suited for tasks were a wide variety of relations are present and data is sparse. Lastly, the NAACL2016 dataset was presented to show the effect of crowd sourcing on RE dataset development. Since the focus of the research behind this set laid on crowd sourcing, merely 5 relation classes are included, which are all defined between `Person` and `Location` entities. Therefore, this dataset should not be used for general RE tasks.

3.4. Automatic labelled dataset development

The discussed methods yield satisfactory results and can be easily adapted to new domains, under the assumption that enough domain-dependent labelled training data is available. In practice, this turns out to be a slightly unrealistic assumption. The development of these labelled datasets is a tedious task which requires a lot of time and effort. Although being a simpler task than manually creating extraction

Dataset	**Language**	**Source**	**# Training sentences**	**# Testing sentences**	**# Relation classes**
SemEval-2010 Task 8 [36]	English	Web pages	8,000	2,717	10
ACE05 [79]	English Chinese Arabic	Weblogs, broadcast news, newsgroups, broadcast conversation	5,958	1,147	7
Altered StanfordRE [49]	English	KBP 2010 and 2013[a] Wikipedia	7,320	1,830	41
NYT [68]	English	New York Times	4,700	1,950	58
NAACL2016 [52]	English	StanfordRE dataset	18,128	164	5

Table 3.1: Properties of closed relation extraction datasets.

[a] https://catalog.ldc.upenn.edu/LDC2018T03

rules, we can conclude that the manual labour has shifted from creating rules to labelling datasets.

Naturally, methods were proposed to reduce the manual work necessary for creating labelled training sets for closed RE. These methods are often referred to as weakly-supervised methods. We discuss two different approaches; bootstrapping and distant supervision. Considering the fact that our main motivation for looking at these methods is to find useful techniques for later use, we only provide a brief historical overview. The best performing approaches will be described in more detail.

3.4.1. Bootstrapping

The general idea behind bootstrapping is that we expand an initial set of labelled examples for each relation, these examples are called the seed instances. First, sentences in which these seed instances occur are retrieved from a text corpus. These sentences are then used to create new extraction patterns, which are used to find and label new instances of the relations in the given text corpus. The newly found instances are added to the seed instances and this process is repeated until a stop criterion is met. In general, bootstrapping techniques deal with a major problem, being semantic drift. Batista et al. [9] define semantic drift as *“the progressive deviation of the semantics for the extracted relationships from the semantics of the seed relationships”*. This means that with each iteration, the semantic meaning of the newly found relationships drifts further away from the seed relationships, which are the only relationships known to be valid. The proposed bootstrapping methodologies all deal with this problem of semantic drift in different ways.

Brin [12] developed a system named Dual Iterative Pattern Relation Expansion (DIPRE). Within DIPRE, the sentences in which the seed relations occur are modelled as three separate strings; words before the first entity (BEF), words between both entities (BET) and words after the second entity (AFT). Using these representations, extraction patterns are extracted by grouping via string matching. Semantic drift is restricted by putting a limitation on the number of instances that can be extracted using a generated pattern.

Agichtein et al. [1] present a DIPRE-inspired system called Snowball. Just like DIPRE, a BEF, BET and AFT string are collected for each seed occurrence. In Snowball, these context strings are represented using a Term Frequency-Inverse Document Frequency (TF-IDF) representation. A similarity measure based on the cosine distances of these three TF-IDF vector representations is used to create a clustering of the seed sentences. The centroids of the clusters form the extraction patterns. The text is scanned again, if the similarity measure between a sentence and a cluster centroid is above a certain threshold, this sentence is extracted and added to the seed occurrences. To control the semantic drift, Snowball uses a scoring mechanism for the extracted patterns and seed instances. Only the instances that score above a certain threshold will be added to the seed instances.

The string and TF-IDF vector representations used in previous models both have a clear limitation. The similarity between two strings or TF-IDF vectors of two phrases is only positive if the phrases share at least one word. For example Batista et al. [9] show that the TF-IDF vectors of *“was founded by”* and *“is the co-founder of”* have a negative similarity, while their semantic meaning is clearly highly similar. To counteract this limitation, Batista et al. propose BREDS, a bootstrapping model based on word embeddings. Please recall that word embeddings are used to represent textual words as numerical vectors. The main property of these embeddings is that words that often appear in a similar context are encoded in similar vectors. So for example the vectors for the *founder* examples will be highly similar, since they often appear in similar textual contexts.

Identical to Snowball, BREDS starts by scanning a document collection for co-occurrences within a sentence of both entities of a seed relationship. The phrases that are found are again represented via a BEF, BET and AFT vector. In contrast to Snowball, BREDS uses word embeddings and custom optimizations to create these vectors. Once the instance vectors are extracted, a single-pass clustering algorithm is used to generate extraction patterns. The algorithm iterates through the extracted instances, computing the similarity between the instance and every cluster using the Snowball similarity measure. The maximum similarity between an instance and each instance in the cluster is returned, given that the majority of similarity scores is higher than a certain threshold. Otherwise, a similarity of 0 is returned. This approach differs from the Snowball approach, where similarities are computed to the cluster centroids. Now, the current instance is assigned to the first cluster whose similarity is greater or equal to a threshold. If all clusters have a similarity lower than the threshold, a new cluster is created. Now that the clusters are created, they are directly used as extraction patterns. The document corpus is rescanned, collecting all phrases containing entity pairs whose semantic types are the same as those of the seed entities. Each of these phrases is encoded using the method described above and the similarity towards all clusters are computed. Whenever the confidence score of an instance and its most similar cluster is higher than a certain threshold, it is added to the seed instances. The writers use this threshold to limit the semantic drift.

In their evaluations, Batista et al. show that BREDS is indeed an improvement on Snowball. Their main improvement is expressed in much higher recall scores. The researchers argue that this is caused by the relaxed semantic matching which is a result of using word embeddings.

3.4.2. Distant supervision

Even though Batista et al. [9] were able to greatly improve the recall of bootstrapping, yielding satisfactory recall scores does remain a problem of bootstrapping methods. This is due to the fact that bootstrapping is dependent on the quality and completeness of the seed instances. If these seeds only cover a part of the possible relations, it is highly unlikely that bootstrapping will find instances of all relations. In this case, the seed instances are too specific and the final recall will be low. Moreover, due to its iterative nature, errors are propagated over iterations and bootstrapping cannot be used at all when there are no seed instances available.

Paradigm introduction

To loosen these restrictions, while still providing an automatically labelled dataset, Mintz et al. [59] introduced the distant supervision paradigm. Distant supervision seeks to link a knowledge base to a corpus of text to create a labelled dataset which can be used for supervised relation extraction. The initial distant supervision proposal relied on one assumption;

> if two entities participate in a known knowledge base relation, **any** sentence that contains both entities is likely to express that relation.

Consequently, if a sentence contains a pair of entities that participate in a known knowledge base relation, features are extracted from that sentence and these features are added to the feature vector for that relation. These features can then be used to train a classifier for the relation extraction task.

Mintz et al. [59] provide a clarifying example. Consider the *location-contains* relation and imagine that the knowledge base holds two instances of this relation; `<Virginia, Richmond>` and `<France, Nantes>`. If sentences *"Richmond, the capital of Virginia"* and *"Henry's Edict of Nantes helped the protestants of France"* are encountered in the corpus, extracted features from these sentences would be added to the feature vector of the *location-contains* example. Some features, like those from the first sentence, would be very useful. Others, like those from the second sentence, would be less useful. If during testing we came across the sentence *"Vienna, the capital of Austria"*, its features would match those of the *Richmond* sentence, thereby providing evidence that `<Austria, Vienna>` has a *location-contains* relation. Using both automatic and manual evaluation, the researchers conclude that distant supervision is able to extract high-precision patterns.

Relaxed distant supervision assumption

Looking at the initial proposal, one might raise legitimate doubt about the validity of its assumption. We already saw that the assumption did not hold in the *Nantes* example and it seems plausible that these cases occur often in internet texts. This is confirmed by Riedel et al. [68], who show that the assumption is violated in approximately 31% of the cases when the Freebase[3] knowledge base is aligned with the New York Times corpus[4]. Therefore, methods have been proposed to relax the distant supervision assumption in two ways. Riedel et al. propose the *expressed-at-least-once* assumption;

> if two entities participate in a known knowledge base relation, **at least one** sentence that contains both entities is likely to express that relation.

Although this assumption trivially holds with more certainty, it complicates the distant supervision task. To solve this task, the researchers introduce a novel graphical model that predicts relations between

[3] `https://developers.google.com/freebase`

[4] `https://catalog.ldc.upenn.edu/LDC2008T19`

entities and which phrases express these relations. As a second contribution they frame distant supervision as an instance of constraint-driven semi-supervision and use SampleRank, a large factor graph discriminative learning model [84], with the *expressed-at-least-once* assumption injected through a truth function. In their evaluations, Riedel et al. show that this relaxed assumption indeed significantly improves the precision of the extracted dataset. In general, this task can be framed as a Multiple-Instance Learning (MIL) problem, for which many solutions are proposed [15]. All sentences in which two entities from a knowledge base relation occur are placed in a so called bag. We know that the truth label for that bag is the specific knowledge base relation, we do however not know the truth labels of the individual instances within the bag.

Surdeanu et al. [76] state a second challenge for distant supervision, showing that the same pair of entities can have multiple semantic relationships between them. This occurs for 7.5% of the entity tuples in the Riedel et al. [68] dataset. As an example, the entity pair `<Alan Turing, England>` can be labelled by both the `BornIn` and `DiedIn` relation. Accordingly, the writers argue that the problem should be tackled as a Multi-Instance Multi-Label (MIML) learning problem. Surdeanu et al. propose a novel graphical model that jointly faces the multiple instances and multiple labels assumptions in a Bayesian framework. The evaluations again show the merit of the proposed model.

Jiang et al. [41] show how the MIML concept can be integrated in modern CNN-based relation extractors. The *expressed-at-least-once* assumption is incorporated by applying a cross-sentence max-pooling layer after the separate sentence-level representations are formed. Thereby, the most significant features over multiple sentences are aggregated. The multi-label problem is tackled by introducing various multi-label loss functions for the classifier. Experiments show that these techniques consistently improve the performance of distantly supervised CNNs.

Noise reduction
Given the automatically generated nature of distant supervision datasets, they often contain noise in the form of wrongly labelled sentences. Qin et al. [65] introduce an adversarial pipeline for distant supervision. Given such a noisy dataset, the generator tries to find valid samples. These generated samples are regarded as negative samples to train a discriminator network. Consequently, the performance of the discriminator decreases as the generator discovers more valid samples. Therefore the generator is trained to maximize the classification error of the discriminator. Such a model is often referred to a Generative Adversarial Network (GAN). Experiments show that this GAN is successful in reducing noise in distantly supervised datasets, which leads to better classification results.

Another noise reduction approach for distant supervision is proposed by Lin et al. [50]. First, a sentence-level feature vector is produced for each sentence of an entity pair using a DNN approach. Next, these sentence vectors are weighted in order to de-emphasize noisy sentences. These weights are automatically optimized during training. This approach is called selective attention.

3.5. Conclusion

In this chapter, we observed that closed relation extraction is an active research domain. We discussed the transition from hand-crafted rules, to hand-crafted features and later to deep learning approaches. We identified that understanding the reasoning behind the deep learning models is an unexplored field within the RE domain, which could be due to the missing necessity of explaining decisions within this domain. However, since we do expect that this could lead to interesting insights and improvements, we suggest it as a topic for future work. Within the state-of-the-art models, CNN- and RNN-based approaches are thoroughly explored, resulting in the observation that they provide complementary information for the RE task. Adversarial networks, regularization and pre-training can be used to further boost the performance. Furthermore, we concluded that valid datasets for testing and training are sparse and found that distant supervision can offer a solution to this problem. MIML, GANs and selective attention improve the usability of these automatically developed datasets. These conclusions summarize our extensive answer to our first research question; What is the state-of-the-art of closed relation extraction?

Even though satisfactory performance can be achieved using the discussed techniques, their closed nature forms a limitation for their applicability in practice. From the definition provided in Section 2.2, we know that all possible relation classes should be known if we want to train or use a closed relation extractor. Moreover, a closed RE system that is trained on a certain set of relations can not be directly used on a disparate relation set. For many use cases, such as heterogeneous internet texts, we do not know exactly which relations we are looking for in advance. Therefore, the next chapter will be focussed on open relation extraction, were the relations are directly inferred from the text without the need to define them up front. Nonetheless, many techniques that were proposed in the closed setting prove to be beneficial in the open setting as well.

4

Open Relation Extraction

The open relation extraction paradigm was introduced to offer a more generic solution for extracting relations from text corpora. In this chapter, we answer our second research question; What is the state-of-the-art of open relation extraction? To that end, we comprehensively discuss this paradigm, describing its history as well as the state-of-the-art approaches. We also describe various post-processing efforts and look at the problem of evaluating open relation extractors. We find that the open RE literature is considerably less voluminous than its closed counterpart.

4.1. Sequence-to-sequence classification

The paradigm of open relation extraction, which is often referred to as Open Information Extraction (OIE), was first defined by Banko et al. [7] as “a new extraction paradigm where the system makes a single data-driven pass over its corpus and extracts a large set of relational tuples without requiring any human input”. Accordingly, methods within this paradigm are expected to provide a better fit for the diversity and size of the web than methods from the closed domain. The main difference we observe between both domains is that the open RE task is typically approached as a sequence-to-sequence task, whereas the closed RE task is approached as a multi-class classification task. In this sequence-to-sequence task, the output vocabulary equals the input vocabulary plus the entity and relation tags. Recall the example from Section 2.2, here the sequence *“Turing was born in England”* forms the input and “<arg$_1$>*Turing* </arg$_1$> <rel> *was born in* </rel> <arg$_2$> *England* </arg$_2$>” could form the output. From the output sequence, we can easily construct a relation tuple as shown in Section 2.2.

Banko et al. [7] presented TextRunner as a first fully implemented open relation extractor. Prior to full-scale relation extraction, a set of relation-independent syntactic patterns is used to automatically extract a labelled training set from a small sample of the input corpus. From this training set, lexical features (as described in Section 3.1.2) are extracted and a label sequencing classifier is trained. During classification, all words between two entities in each corpus sentence are extracted as a potential relation and post-processing steps are applied to clean up these potential relations. Then, the classifier is used to label each token from each potential relation as being part of the relation or not. In a final step, TextRunner merges tuples where both entities and the normalized relation are the same and counts the number of sentences in which each extraction was found to determine a confidence score for that tuple.

4.2. Improvements

From its design, we can derive that TextRunner is highly dependent on the relation-independent syntactic patterns that are used to extract the training set. As a result, TextRunner can only find relations that match a pre-defined pattern. Defining these patterns is not a trivial task and each language requires its own set of syntactic patterns.

4.2.1. Conventional improvements

To relax this requirement for pre-defined patterns, Wu et al. [86] seek to automatically extract a training set in a way that is highly similar to distant supervision (discussed in Section 3.4.2). In a system they call Wikipedia-based Open Extractor (WOE), the writers automatically match information from a Wikipedia infobox[1] with its corresponding text to find positive training instances. As a second alteration, Wu et al. include syntactic features next to the lexical features (both are discussed in Section 3.1.2). Although both alterations significantly boost the performance, the syntactic features add an extra dependency on external NLP tools and slow down the model by approximately 30 times [86]. Given the fact that Wikipedia currently holds 140 languages that have 10,000 or more articles[2], WOE could potentially be used in a multilingual fashion if language-independent features were to be used.

In contrast, many proposed open RE improvements focus specifically on exploiting language structures that are specific to the English language. Fader et al. [27] introduce ReVerb, a system that adds a syntactic and a lexical constraint for the relation phrases. These constraints were the result of a thorough linguistic analysis of randomly sampled English sentences. They significantly improve the performance of the relation extractor, at the cost of making it even more limited towards the English language. Similarly, Del Corro et al. [24] present ClausIE, a system that "exploits linguistic knowledge about the grammar of the English language, to first detect clauses in an input sentence and to subsequently identify the type of each clause". It differs from ReVerb in the sense that it can be directly applied, no labelled or unlabelled training data is needed. In a similar fashion, Angeli et al. [4] pre-process a sentence using linguistic knowledge to produce coherent clauses.

Mausam et al. [56] identify two weaknesses of WOE and ReVerb. First, they only extract relations that are mediated by verbs. Second, they ignore context, which could result in extracting tuples that are not asserted as factual. The writers present OLLIE to address both weaknesses. By identifying relations mediated by nouns and adjectives, they expand the syntactic scope of their system. Furthermore, they analyse the context around an extracted relation to find cases were the extraction is hypothetical or conditionally true.

4.2.2. Unsupervised open relation extraction

Elsahar et al. [26] propose an interesting procedure for open RE, which does not require any labelled data; unsupervised open relation extraction. It deviates from the before-mentioned techniques in the fact that Elsahar et al. frame the open RE task as an unsupervised clustering problem instead of a sequence-to-sequence classification problem. As an output of their system, they want to create clusters of sentences that entail similar relations, without knowing what these relations are. For this purpose, their system requires the word embeddings, relation entity types and dependency tree of the input sentence. Just as we have seen in closed RE systems, sentence-level features need to be extracted from the individual word embeddings. Since there is no training data, we can not train a DNN for this purpose. Elsahar et al. present a novel method which calculates a weighted average of the word embeddings to form the sentence-level embedding. The weights are calculated by checking if a word lays on the path between the two relation entities in the dependency tree. This sentence level representation is concatenated with the relation entity types to form the feature vector. DNNs implicitly reduce the number of features. To mimic this feature reduction step in the unsupervised setting, the writers apply Principal Component Analysis (PCA). Finally, Hierarchical Agglomerative Clustering (HAC) is applied to form the output clusters. The notable benefit of this method is that no training data is needed. That being said, the entity tagger and dependency parser are trained on labelled data and should be available and performant for this method to work. Moreover, from the output we can only derive that certain sentences entail similar relations, we can not directly extract relation tuples.

4.2.3. Neural open relation extraction

The discussed methods use lexical and/or syntactic features. In Section 3.2, we discussed arguments to move away from these features for the closed RE task. The same arguments can be used to make a similar case for the open RE task. We want to move away from these features, since they cause a dependency on external NLP tools, which results in error propagation. Moreover, the vast majority

[1] A set of tuples summarizing the key information on a Wikipedia page.

[2] `https://en.wikipedia.org/wiki/List_of_Wikipedias`

of these NLP tools only work for the English language, hindering the extension to new languages. The same can be said for the improvements discussed above, which specifically exploit knowledge about the English grammar. Accordingly, three recent research efforts were made to apply deep neural networks for the open RE task, an approach that led to state-of-the-art results in the closed paradigm (see Section 3.2). We can not directly apply the closed DNN architectures, because the problem is not longer a multi-class classification problem but a sequence-to-sequence problem. Therefore, new methodologies are needed. The three mentioned DNN open relation extractors offer three different perspectives on the sequence-to-sequence task.

Cui et al. [20] propose a novel technique to tackle the sequence-to-sequence RE problem, using an encoder-decoder framework. A graphical representation of this system is presented in Figure 4.1. The system first embeds the input sentence in word embeddings. These word embeddings are encoded into hidden representations by the encoder, which uses an LSTM. An attention method which weights these hidden representations is applied, before they are used as the input for the decoder. The decoder also uses an LSTM. It tries to decode the hidden representation into the correct tagged output sentence, as shown in the example in Section 4.1. Since the output vocabulary is directly taken from the input vocabulary (expect for the tags), a copying mechanism is applied before the decoder finds its final output. Ciu et al. train the model using a training set that they create by extracting relation tuples from Wikipedia using an existing open relation extractor. They only select tuples that receive a confidence score higher than 0.9 to create this training set. This approach to automatically create a labelled dataset differs from the approaches in the closed domain, discussed in Section 3.4.2, but can be seen as a bootstrapping approach.

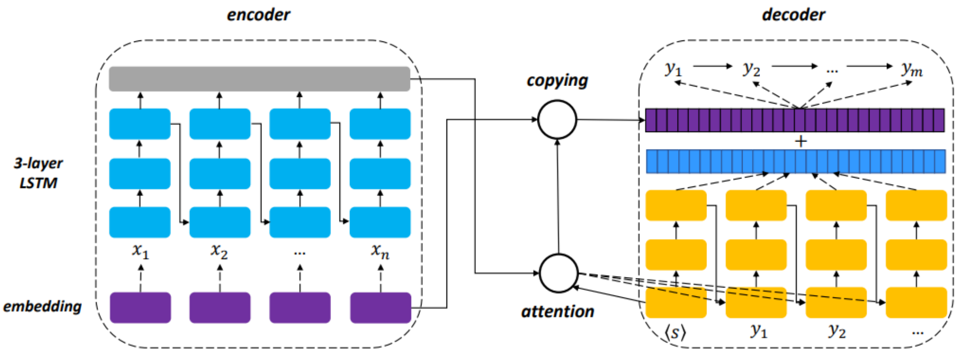


Figure 4.1: Open RE encoder-decoder framework as depicted by Ciu et al. [20].

Stanovsky et al. [75] present multiple contributions to the open RE task. First, they introduce a custom tagging scheme which contains more information than the simple markers used by Cui et al. [20], we will refer to this scheme as Stanovsky tagging. This tagging scheme can best be explained using an illustrative example. Given the sentence

Alan Turing, a computer science pioneer, was born in England.

it will be tagged as follows;

$Alan_{A0-B}$ $Turing_{A0-I}$ $,_{O}$ a_{A0-B} $computer_{A0-I}$ $science_{A0-I}$ $pioneer_{A0-I}$ $,_{O}$ was_{P-B} $born_{P-I}$ in_{P-I} $England_{A1-B}$ $._{O}$,

where B represents the beginning, I the inside and O the outside of a relation tuple argument, A_i represents the i^{th} relation entity (arguments) and P represent the relation (predicates) between these entities. Second, they create and publish a sizeable open RE dataset, which is labelled using the introduced tagging scheme. Third, they propose a novel, RNN-based open relation extractor. The model is depicted in Figure 4.2.

This relation extractor makes use of both the tagging scheme and the new dataset. Additionally, this system makes use of word embeddings and a PoS-tagger. It works by first finding candidate relation heads (i.e. P-B) using the PoS-tagger and normalization techniques. Then it creates a feature vector for each word in the input sentence and each candidate relation head by concatenating the embeddings of the current word, the PoS-tag of the current word, the current candidate relation head and its PoS-tag. These feature vectors are used as the input of a bi-directional LSTM, which is linked to a softmax classifier to predict the Stanovsky tag for each word. Effectively turning the sequence-to-sequence problem into a multi-class classification problem for each word, while taking information from the whole sentence into account. From a tagged sentence, we can infer a relation tuple. The confidence scores for these tuples can be calculated by multiplying the probabilities of the B and I labels participating in the relation[3].

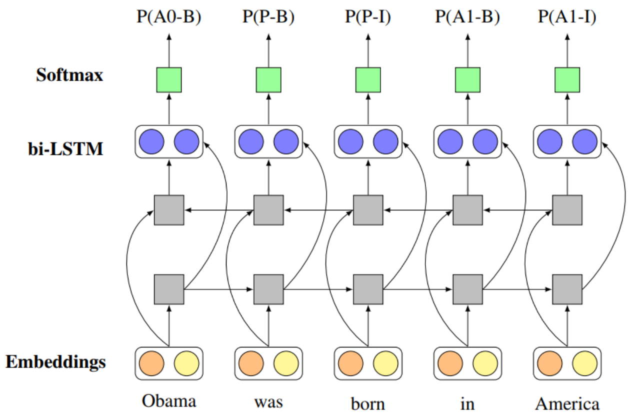


Figure 4.2: Open RE RNN framework as depicted by Stanovsky et al. [75].

Jia et al. [39] propose multiple improvements on neural open RE, which result in the current state-of-the-art model. First, they create a training set which is significantly more extensive then previously available datasets. This set is created by extracting relation tuples from a large news corpus using three existing relation extractors. If a tuple is extracted by all three extractors, it is added to the training set. Second, Jia et al. extend the Stanovsky tagging scheme by introducing E and S tags for end and single tuple arguments, respectively[4]. The example sentence will be tagged as follows;

$$Alan_{A0-B}\ Turing_{A0-E}\ ,_{O}\ a_{A0-B}\ computer_{A0-I}\ science_{A0-I}\ pioneer_{A0-E}\ ,_{O}\ was_{P-B}\ born_{P-I}\ in_{P-E}\ England_{A1-S}\ ._{O}.$$

This scheme has proven to be more expressive for related NLP tasks [67]. Third, the writers present a novel, state-of-the-art open RE model. This model is named the Neural Sequence Tagging (NST) model, it is depicted in Figure 4.3.

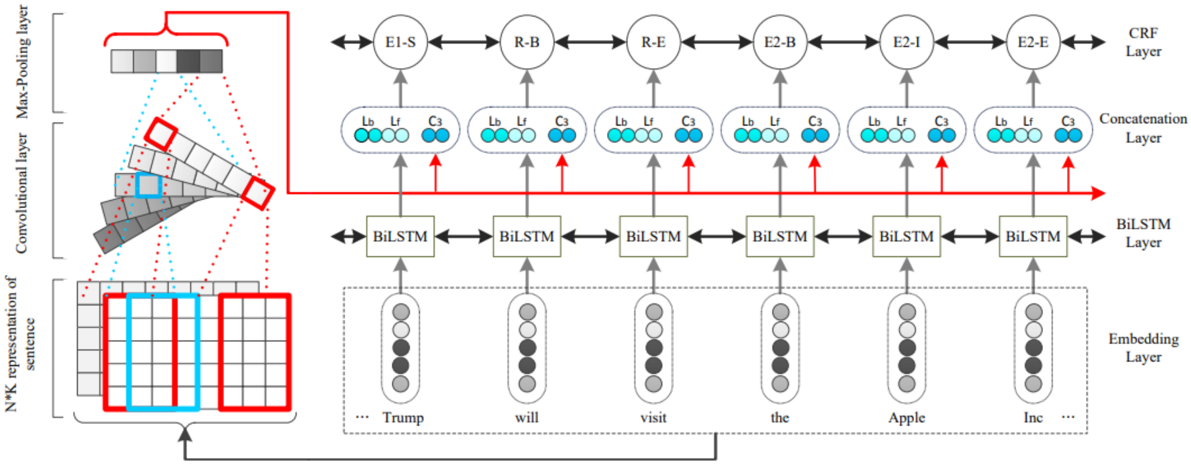


Figure 4.3: NST framework as depicted by Jia et al. [39].

[3] This proved to be the best combination heuristic according to Stanovsky et al. [75].

[4] Also, A and P tags are replaced by E and R tags respectively. This is solely a cosmetic alteration.

As was discussed in Section 3.2.3, CNNs and RNNs provide complementary information for the RE task. Relational words tend to occur in the neighbourhoods of entities. Therefore, certain parts of the input sentence might have a higher chance of containing relation words than others. A CNN is used to capture this local feature information from the input sentence. At the same time, a bidirectional LSTM is used to capture the forward and backward context of each word. The output of the CNN and the bidirectional LSTM are concatenated and a CRF is used to make the final prediction on the predicted tag for each word. A CRF is used because it can efficiently consider correlations between past and future tags to predict the current tag. As we concluded several times, a multilingual approach preferably does not depend on external NLP tools. Although the best performing model that was presented makes use of a PoS-tagger, results are also presented for a model that only uses word embeddings. Even though this leads to a 5.2% decrease of the F_1-score, the performance remains satisfactory.

4.3. Post-processing

We have seen that the output relation tuples of open relation extractors typically directly extract words from the input sentences. Therefore, two relations that are semantically the same could end up in different relation tuples. For example, the sentences *“Alan Turing lives in England”* and *“Alan Turing resides in England”* lead to `<Alan Turing, lives in, England>` and `<Alan Turing, resides in, England>`, respectively. To correct for these differences and to improve the usability of the extracted relation tuples in other ways, post-processing steps were presented for the open RE domain.

An instinctive approach to generalize the found relation tuples, is clustering those relations and entities that are (near) synonyms, such as *lives in* and *resides in* or *Turing* and *Alan Turing*. Yates et al. [90] present an unsupervised system which does exactly that; Resolver. This system uses string similarity to cluster similar relations and entities. More modern techniques for disambiguating open RE tuples have also been presented. For instance, Sanchez et al. [71] adopt word embeddings to find similar relation and entity expressions.

If we take this idea one step further, we could try to form a graph out of the extracted relations, after a disambiguation step. Levy et al. [48] propose a technique which creates these so called entailment graphs from open RE tuples. Such a graph does not only cluster equivalent expressions, it also depicts generalization relations (e.g. *Scientist* is a generalization of *Turing*). Creating these entailment graphs is a research domain on its own, which falls outside of the scope of this work. It is nonetheless a very interesting step that can greatly improve the usability of the extracted relations in many real-world scenarios.

Soderland et al. [72] display a method that takes the extracted open relation tuples as input and produces relation instances grounded in a given ontology. This could for example be useful in the scenario that we notice a lot of relations from the same domain when we manually inspect samples from the extracted relation tuples. If an ontology is available for that domain, we can map the open relation tuples to this ontology to find more general relations.

4.4. Datasets

The initial open relation extractors were trained on data that was extracted by hand-crafted linguistic extractions of a sample of the target dataset or Wikipedia infobox linkages [7, 86], these approaches can be seen as types of distant supervision (as discussed in Section 3.4.2). Although the majority of these works make their dataset freely available, they are too small to be considered for modern day approaches. Therefore, we will not further discuss these datasets. More recent works use larger, supervised open RE datasets, which are either automatically created using a bootstrap-like approach or derived from datasets of related NLP tasks. These datasets are summarized in Table 4.1, all datasets in this table are comprised of English sentences. It is important to note that the state-of-the-art neural methods require significantly more data than conventional approaches. This explains why the datasets grew by several orders of magnitude as new neural methods were introduced. Notwithstanding the fact that size is a very important factor, it is not the only property which influences the final performance of the relation extractor. Since these datasets were created using different methods, they all contain different levels of noise, which influences the performance of the final classifier. For this reason, multiple training sets can be examined to derive which of these provide the best fit to the model at hand.

Dataset	Domain	Creation method	# Sentences	# Tuples
NeuralOpenIE [20]	Wikipedia	High confidence extractions of one existing RE method	1,597,830	1,597,830
NSL4OIE [39]	News	Consensus of three existing RE methods	395,715	477,701
AW-OIE [75]	Wikipedia and news	Derived from question answering dataset	3,300	17,165
OIE2016 [74]	Wikipedia and news	Derived from question answering dataset	3,200	10,359

Table 4.1: Properties of open relation extraction datasets.

The increased data need is less of an issue for testing. Although a bigger test set will typically lead to evaluation metrics that better reflect the real-world performance of the model, the need for more test data does generally not depend on the model. Additionally, factors like the representativeness of the test data for the data on which the model will be used determine how well the evaluation metrics represent real-world performance. In open RE literature, we see that most works evaluate their model by either putting aside a part of the training set for testing or by using a different test set. Although the datasets of the initial works do not suffice for training the neural models, they are repeatedly used for testing these models. In the open domain, the representativeness of the test data for real-world data can lead to difficulties. Please remember the definition of the recall given in Equation 2.2. Computing this evaluation score requires one to specify *all* relations in the test corpus. If we look at the datasets in Table 4.1, we can conclude that the datasets that were created by using existing relation extractors are probably not very suitable for testing. Since only high confidence extractions are used, it is likely that only a subset of all relations in the corpus are found[5]. Consequently, the recall scores found when using these sets for testing are probably too optimistic.

In order to more realistically evaluate open RE models, Bronzi et al. [13] present a large-scale evaluation system that makes use of an external knowledge base and a web search engine. To measure the true precision and recall of a system, the writers show that we need to estimate the size of four different evaluation regions, depicted as *a, b, c* and *d* in Figure 4.4. In this figure, the extractions made by the RE system are represented by **S**, the relations in the external knowledge base by **D** and the ground-truth relations that appear in the test corpus by **G**. Furthermore, *a* contains correctly extracted relations from the input which are not in the knowledge base, *b* is the intersection of relations that are both in the system output and in the knowledge base[6], *c* contains relations in the test corpus which are not in the system output and *d* contains relations in the test corpus which are neither in the knowledge base nor in the system output.

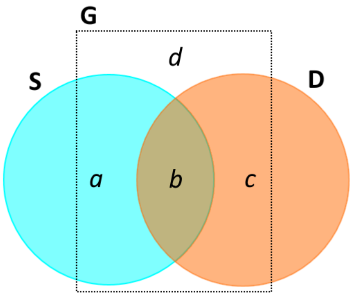


Figure 4.4: Graphical representation of evaluation regions.

Bronzi et al. observe that regions *a* and *b* form all true positives, while regions *c* and *d* form all true negatives. Since regions *a, b, c* and *d* form **G**, the writers redefine the precision as

$$P = \frac{|a| + |b|}{|\mathbf{S}|} \tag{4.1}$$

[5] If all relations are found, the existing relation extractor(s) would yield a perfect recall score of 1, which is highly unlikely.

[6] Bronzi et al. assume that this region is composed of correct facts only, since it is unlikely that an incorrect relation is both extracted by the system and in the knowledge base.

and the recall as

$$R = \frac{|a| + |b|}{|a| + |b| + |c| + |d|}. \tag{4.2}$$

First, $|b|$ is determined. To this end, the evaluation system uses a matching algorithm to try and match the relations extracted by the relation extractor with the relations present in the knowledge base. Hereafter, $|a|$ is approximated using Pointwise Mutual Information (PMI) on web documents. The PMI is calculated by dividing the number of documents returned by query "$\texttt{Entity}_1$ AND `Relation` AND $\texttt{Entity}_2$" by the number of documents returned by query "$\texttt{Entity}_1$ AND $\texttt{Entity}_2$". Relation tuples with a higher PMI are assumed to have a higher chance of being correct. A relation is believed to be correct if the PMI is above a certain threshold. By calculating the PMI of the relations extracted by the relation extractor which are not in *b*, we can estimate $|a|$. In order to calculate regions *c* and *d*, a new set **G'** is defined, which is a superset of the ground truth **G**. The idea is to first form **G'** and then approximate **G** by removing incorrect relations using the knowledge base and PMI. **G'** is produced by adding combinations of extracted entities and relations from the test corpus. Naturally, this leads to a very big set. Bronzi et al. reduce this size by applying heuristics, they for example only consider entities from the same sentence. Once **G'** is produced $|b| + |c|$ is estimated by counting matches between **G'** and **D**. Then, $|c|$ can be estimated by subtracting $|b|$ from the number of matches. Finally, PMI is applied to estimate the correct relations which are in **G'**, but not in **D**. From this, $|d|$ can be computed by subtracting $|a|$. Experiments show that results presented for two conventional open relation extractors might indeed be too optimistic.

Even though this method enables large-scale open RE evaluation on many different text corpora, there are some remarks that need to be made. Looking at the creation process of **G'**, it seems likely that |**G'**| is larger than |**G**|. As a result, the recall scores might be a bit pessimistic compared to the actual recall of the system. Furthermore, this method assumes that the relations that are extracted from the test corpus are public knowledge, since they should be available in either a knowledge base or web search corpus to be seen as valid. When the test corpus contains private knowledge (e.g. emails or social media data), this method will not yield trustworthy evaluation results. Moreover, an extensive web search corpus and knowledge base should be available in the test corpus language. For English, these sources are available. Nonetheless, this method will not be suited for every language.

4.5. Conclusion

We discussed the relatively new field of open RE, in which considerably less methods were proposed compared to the closed paradigm. From its introduction, the conventional improvements mainly focussed on exploiting structures in the English language. Also, limited efforts were made to tackle this problem in an unsupervised fashion. In recent years, the focus increasingly shifted towards supervised neural approaches. Three neural models were discussed, which led to state-of-the-art results in the field. These models indicated that we can approach the open RE problem as a sequence-to-sequence generation or a sequence classification task. Moreover, we have seen that, although NLP tools such as PoS-taggers can improve the performance, only using word embeddings can lead to satisfactory results. Post-processing steps were shortly touched upon to give an indication of the research that is conducted in that field, which falls outside of the scope of this work. Finally, we discussed that one should be careful during the evaluation of the discussed models, since test sets might not reflect the true precision and recall of the tested system. To yield more trustworthy evaluation results, a tailored evaluation framework was discussed. This chapter provides an answer to our second research question; What is the state-of-the-art of open relation extraction?

Considering the main research topic of this work, multilingual open relation extraction, the state-of-the-art neural approaches discussed in this chapter offer a solid starting point for our research. In Chapter 6, we will propose alterations aimed at applying these models in a language-consistent setting. Moreover, improvements that were made in the closed domain will be applied to these models to further improve their performance. Before we dive into our proposed model, we discuss multilingual RE literature in the next chapter, to get an idea of the best practises in that field.

5

Multilingual Relation Extraction

We turn our attention towards our third research question; What is the state-of-the-art of multilingual relation extraction? We start by identifying two general techniques which make relation extraction available for multiple languages. From there, we discuss state-of-the-art instantiations of both techniques and their training and evaluation processes. Finally, we succinctly discuss word embeddings for multilingual NLP.

5.1. Multilingual relation extractors

In previous chapters, we discussed an extensive range of both closed and open RE systems. We have seen that the vast majority of these models is trained and tested on English datasets. If these models are applied in a multilingual setting, we encounter two weaknesses. First, the vast majority of these systems use external NLP tools such as PoS-taggers and dependency parsers. These models need to be adapted to use tools for the given language, which is a non-trivial process. Second, this approach would fail to exploit information that is present over multiple languages (language-consistent patterns). Both of these weaknesses are addressed by two different multilingual RE techniques; cross-lingual RE and language-consistent RE. Cross-lingual systems try to extract relations from a source language by exploiting information and systems from a target language, thereby removing the need for a labelled training set or NLP tools in the source language. On the other hand, language-consistent systems exploit information that is present in multiple languages.

5.1.1. Cross-lingual relation extraction

Cross-lingual relation extraction is defined as “the task of extracting relations from the source language and representing them in the target language” [97]. In a typical use-case of such a system, the user has no knowledge about the source language, which is why he wants to extract relations in a known language (typically English). An intuitive approach for such a cross-lingual RE system is to first translate the source language into the target language and then utilize an existing relation extractor. This idea is implemented and evaluated by Faruqui et al. [28]. Although this technique is attractive due to its simple nature, it does come with certain limitations. First of all, the writers assume that a performant source to target language translator is available. Naturally, the performance of the relation extractor is highly dependent on the quality of these translations. Even though this might be a valid assumption for popular languages, it might prove to be problematic for low-resource languages. Secondly, cross-lingual models assume that the relation extractor that is used will be applicable to the translated source language, even though it is created for the target language. This assumption could be to strong when the language structures within both languages differs a lot.

In an effort to relax the translator assumption and to tailor the translator to the RE task at hand, Zhang et al. [96] present their joint Machine Translation/Information Extraction (MT/IE) system. Instead of first translating the source text and then applying a relation extractor, they jointly train a machine translation and relation extraction model. The translator assumption is replaced by the assumption that a bi-text corpus (e.g. a corpus of Chinese sentences and their English translations) is available. The PredPatt algorithm [83] is used to tag entities and the relations between them in the target part (i.e. English part) of the corpus. PredPatt uses a set of manually-written patterns to extract predicate-argument structures from texts and can be used as a relation extractor. Please note that any relation extractor that works on the target language could be used in this step. Zhang et al. now approach the cross-lingual RE task as a sequence-to-sequence problem, where the source sentences form the input and the tagged target sentences form the output. Given that the source sentences are embedded using word embeddings before they are used as the input, these word embeddings either need to be available or created for the source language. The <source sentence, tagged target sentence> pairs are used to train an LSTM-based encoder-decoder which is highly similar to the model presented by Cui et al. [20] (as described in Section 4.2.3), minus the copying mechanism (since there is no need to copy the source sentences). Experimental results show significant improvements of the joint approach compared to the pipeline model.

As an extension Zhang et al. [97] simplify the tagging vocabulary to ease the burden of the decoder and they introduce a selective decoding mechanism. Instead of one decoder, multiple decoders are trained. Each decoder learns the conditional probability of decoding a specific type of token. A selector is trained to decide which decoder to use for each word. Zhang et al. present a novel approach to efficiently train this model and their experimental results indicate significant improvements over their original model described above. That being said, these models still have a set of limitations. The model assumes that a bi-text is available between English and the source language. Although this is plausible for many languages, it might become problematic for low-resource languages. Manually creating these bi-texts is a tedious task which requires manual effort by expert linguists. Additionally, these models fail to explicitly exploit information that is consistent over languages.

5.1.2. Language-consistent relation extraction

To remove the dependency on bi-text corpora or translators and to introduce a mechanism for exploiting information that is consistent over languages, language-consistent relation extraction was proposed. Language-consistent RE literature assumes that relation patterns in sentences are substantially consistent between different languages. This assumption can be exploited to train a single model which gathers information from multiple languages. In the previous section, we have seen that cross-lingual relation extractors exist for the open RE domain. In contrast, language-consistent relation extractors are currently solely proposed for the closed domain.

Lin et al. [51] present a language-consistent closed neural relation extractor, called Multilingual Neural Relation Extraction (MNRE). Their system first embeds the words in each sentence from a corpus that contains texts in different languages, using word and position embeddings. Again, these embeddings should be either available or created for the used languages. Next, a sentence-level representation is extracted using a PCNN (as discussed in Section 3.2.3). To exploit information from within, as well as from between languages a novel attention mechanism is proposed. Here, monolingual attention selects informative sentences within a language, while multilingual attention measures pattern consistency among languages. In Section 3.4.2, we succinctly described that a monolingual attention mechanism simply weights each sentence-level representation to end up with a relation representation for each possible relation. These weights are automatically learned during training. Since each language has its own characteristics, different monolingual attentions are adopted for each language in the dataset. Multilingual attention works highly similar. It differs in the fact that the attention weights are jointly trained over all sentences in every language. Therefore, multilingual attention is a generalization of monolingual attention. If only one language would be available in the corpus, both attention weights would be the same. Wikipedia-based distant supervision is employed to create training sets in multiple languages, the attention mechanisms perform a de-noising role in this process. Experiments show that considering pattern consistency among languages indeed significantly boosts the performance on multilingual test sets, validating the assumption of the writers. Moreover, Lin et al. show that their system performs considerably worse if only multilingual attention is used, which would be a truly language-consistent model. They conclude that each language has its specific characteristics to express relation patters, which need to be taken into account in the model.

The model that currently achieves state-of-the-art performance is based on MNRE, but includes several improvements to tackle problems of the MNRE model. In their experiments, Wang et al. [81] find that sentence-level representations in different languages yielded by MNRE are mostly clustered by language in the feature space, as can be seen in Figure 5.2b. We expect the model to extract relation patterns that are consistent over languages (i.e. that are independent of the language) from this space. Naturally, this is quite difficult if the sentence-level representations hardly overlap between languages in this feature space. To alleviate this problem, Wang et al. use not one, but two sentence-level feature extractors for each sentence in a given language. A language-individual feature extractor extracts sentence-level features within the language and a language-consistent feature extractor extracts sentence-level features shared over languages. To ensure that the language-consistent extractor truly extracts language-consistent patterns, an adversarial training strategy is applied. Therefore, this model is named Adversarial Multilingual Neural Relation Extraction (AMNRE), it is depicted in Figure 5.1. The adversarial strategy is highly similar to the strategy applied by the DANN model described in Section 3.2.4. Instead of training a discriminator to classify the domain, the discriminator in AMNRE is trained to classify the language. By maximizing the loss of the discriminator, the sentence-level feature extractor is enforced to focus on language-consistent features. Applying two feature extractors results in two different feature spaces; an individual semantic space per language and a consistent semantic space. Both have their own attention mechanism. Moreover, Wang et al. add a constraint to the model which tries to make the consistent semantic space orthogonal to the individual semantic space, in order to fully split language-individual and language-consistent features over these spaces. Next to common evaluations which show the performance gains of this model, the writers also provide an intuitive graphical visualization of their contributions, which can be found in Figure 5.2. Figure 5.2a shows that the

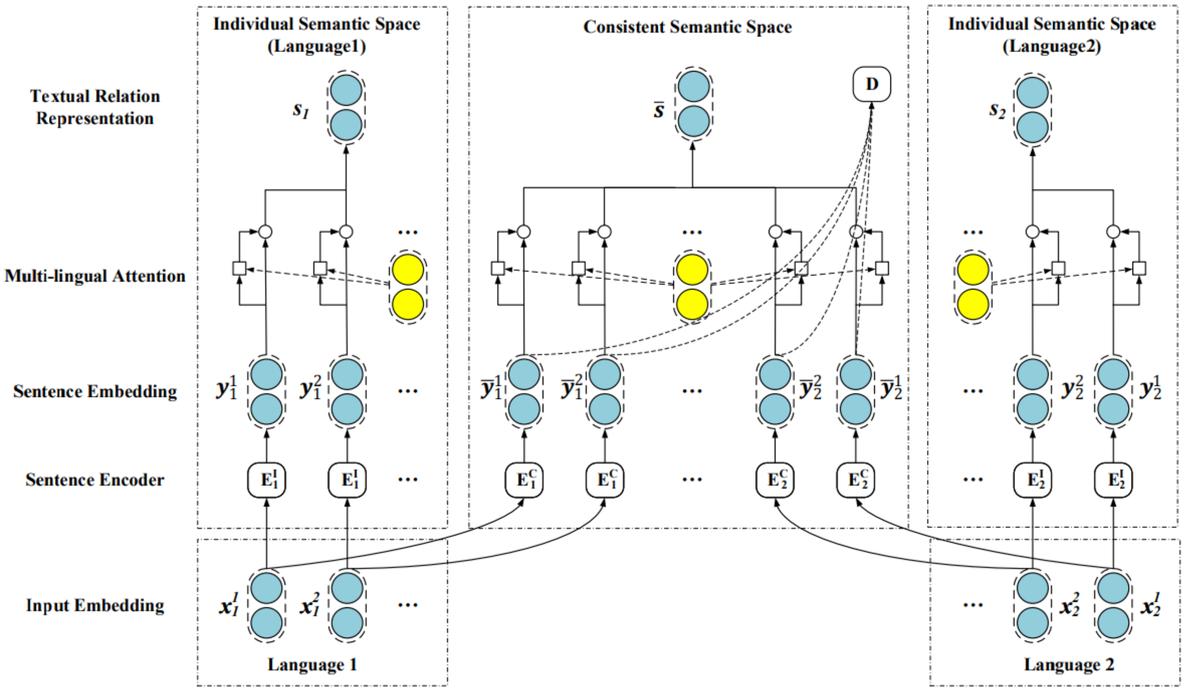
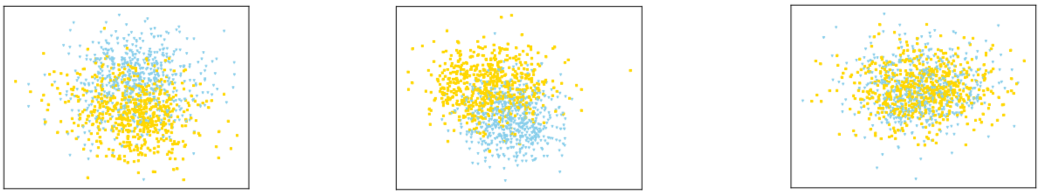


Figure 5.1: AMNRE as depicted by Wang et al. [81] for two languages (it can be extended to more languages).



(a) The same English sentences encoded by the language-individual and language-consistent feature extractor.

(b) English (yellow) and Chinese (blue) sentences encoded by their language-consistent feature extractors *without* adversarial training.

(c) English (yellow) and Chinese (blue) sentences encoded by their language-consistent feature extractors *with* adversarial training.

Figure 5.2: The visualization of sentence-level AMNRE features created using t-SNE, provided by Wang et al. [81].

orthogonality constraint effectively separates the individual and consistent feature spaces[1], since there are clear differences between both sets. Figures 5.2b and 5.2c show the clear advantage of adversarial training. After adversarial training, the sentence-level representations between languages are aligned much better. Additionally, the writers show that sentence-level features of an English sentence and its direct Chinese translation are way more similar than that English sentence and a different English sentence that contains the same entities, as is desired.

Concluding, AMNRE successfully exploits information that is present within, as well as between languages. Yet, this model also comes with a set of limitations, given that it is a closed relation extractor. These limitations were discussed in Section 3.5. Although language-consistent models remove the dependency on translators or bi-text corpora, they do need ORE training data for every language that is added to the model. We hypothesize that language-consistent models could also be used in scenarios were such training data is scarce for a new language, by employing models that are trained on similar languages. This intuition is not yet examined in existing literature and will be part of our experiments in Section 7.4.

5.2. Datasets

In Table 5.1, we provide an overview of datasets used in the multilingual RE domain. Of course, English datasets described in Tables 3.1 and 4.1 can also be used to train or test a multilingual model on the English language. Looking at the limited size of most of these datasets, it becomes clear why distant supervision is often employed to gather training data in this domain. Although the majority of these sets are too small to train a neural model on, they can be used for evaluation purposes. Given the clear superiority in terms of number of languages and sentences of the WMORC$_{auto}$ dataset, this will be our main source of training data.

Dataset	**Languages**	**Source**	**Creation method**	**# Sentences**	**# Tuples**
WMORC$_{\text{auto}}$ [28]	61 languages	Wikipedia	Bootstrapped using existing relation extractor	~100,000 - 9,000,000 per language	~100,000 - 9,000,000 per language
WMORC$_{\text{human}}$ [28]	Russian French Hindi	Wikipedia	Hand-tagged by professional linguists	2,595	2,595
MT/IE [96]	English[2]	GALE project[3]	PredPatt extractions	990,000	990,000
Raw Web [100]	Spanish	Web pages	Hand-tagged by two annotators	159	216
Parallel En-Sp [99]	English Spanish	School text books	Hand-tagged by two annotators	136	276
GerNews [8]	German	News	Automatic annotation using constraints	150	506
GerBH [8]	German	Encyclopaedia	Automatic annotation using constraints	100	452
ItalIE [23]	Italian	Unknown	Hand-tagged by two annotators	240	240
MNRE [51]*	English Chinese	Wikipedia and Baidu Baike	Distant supervision using Wikidata[4]	2,454,966	103,210

Table 5.1: Properties of multilingual relation extraction datasets.
*This is the only closed RE dataset in the table.

[1] No visualization without this constraint is provided, yet experimental results support the same conclusion.

[2] This corpus also contains non-labelled Chinese translations for every labelled English sentence.

[3] https://www.ldc.upenn.edu/collaborations/past-projects/gale/data/gale-pubs

[4] https://www.wikidata.org

5.3. Multilingual word embeddings

During our discussion on the AMNRE model in Section 5.1.2, we saw that performance gains were achieved by aligning the sentence representations of multiple languages in a single space. Many proposals were made to achieve a similar latent consistency among languages on a word level, which came to be known as multilingual word embeddings.

Multilingual word embeddings are created to represent words from multiple languages in a single vector space. Words with similar semantic meaning, both within and between languages, should be close to each other in this space. Mostly, these embeddings are created in a supervised fashion, i.e. bi-text corpora or bilingual dictionaries were used to create these embeddings [2]. Joulin et al. [42] present the state-of-the-art in supervised multilingual word embeddings. Their goal is to learn bilingual mappings between languages, which can be used to link multiple languages together. To that end, they frame this problem as a retrieval task. The multilingual embeddings are obtained by optimizing a retrieval criterion on a bilingual dictionary. As a second contribution, Joulin et al. released pre-trained aligned word embeddings for 44 different languages that can be freely used for a variety of NLP purposes.

Although these supervised multilingual word embeddings can ease the burden of a classifier that works in a multilingual setting, they do require expensive external sources such as bilingual dictionaries. Especially for low-resource languages, this might become a problem. For this reason, multiple techniques to tackle this problem in an unsupervised fashion were proposed. Chen et al. [16] present an unsupervised multilingual word embedding (UMWE) model. This model does not require any bilingual data sources, it only requires a large monolingual corpus (or equivalently, pre-trained word embeddings) for each language. The UMWE model consists of two components, being multilingual adversarial training and multilingual pseudo-supervised refinement. The multilingual adversarial training strategy uses an encoder, decoder and discriminator for each language. For every language, the encoder maps a sample batch of words from the language word embedding space to a shared embedding space. Then, a decoder of a random language is selected to decode the sample batch from the shared space to a language word embedding space. These decoded words, in combination with original word embeddings from the random language, are presented to a discriminator which tries to predicts the likeliness of an embedding belonging to the random language. The encoding-decoding process tries to maximize the loss of the discriminator, i.e. it tries to make the converted vectors look as authentic as possible. This stimulates the model to learn a shared multilingual embedding space. The multilingual adversarial training strategy is depicted in Figure 5.3.

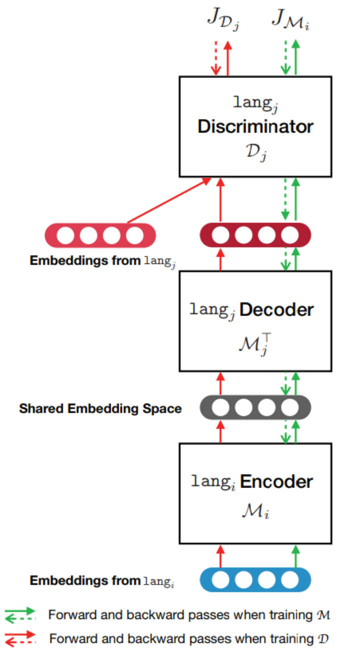

$J_{\mathcal{D}_j}$
$J_{\mathcal{M}_i}$
$\mathtt{lang}_j$ Discriminator $\mathcal{D}_j$
Embeddings from $\mathtt{lang}_j$
$\mathtt{lang}_j$ Decoder $\mathcal{M}_j^\top$
Shared Embedding Space
$\mathtt{lang}_i$ Encoder $\mathcal{M}_i$
Embeddings from $\mathtt{lang}_i$
Forward and backward passes when training $\mathcal{M}$
Forward and backward passes when training $\mathcal{D}$

Figure 5.3: Multilingual adversarial training strategy as depicted by Chen et al. [16].

The multilingual pseudo-supervised refinement method is used to refine the embeddings obtained by adversarial training, especially for infrequent words. This refinement method creates a lexicon for each pair of languages from the mutual nearest neighbours of the most frequent 15,000 words in both languages. The most frequent words are believed to have the best shared space mappings. Now, the two words from two random lexicons are used in an iterative algorithm to minimize the mean square loss between these words in the shared space.

One might wonder why this model is an improvement for low-resource languages, since word embeddings or a large corpus for every language are still a prerequisite for this model. Experiments show that word embeddings for these low-resource languages (e.g. Farsi) are greatly improved. This is accounted to the fact that the initial embeddings for low-resource languages are expected to be noisy. The UMWE model uses information from all other languages in the model to de-noise these low-resource embeddings by mapping them to the shared space.

5.4. Conclusion

In this chapter, we discussed both cross-lingual and language-consistent RE models. We conclude that cross-lingual approaches can be used when we need to extract relations from a source language for which we do not have a labelled training set. We do however need to possess either a performant translator or a sufficiently large bi-text corpus between English and the source language. Moreover, cross-lingual approaches only exploit information from the target language, disregarding patterns that might be specific to the source language or patterns that are consistent over languages. On the other hand, language-consistent approaches can be used to exploit relation patterns that are consistent over multiple languages. This overview provides an answer to research question three; What is the state-of-the-art of multilingual relation extraction?

Given the wide range of languages that is available on the internet, we are more interested in harvesting pattern information from all of these languages, instead of only from a target language. Therefore, we turn our attention towards language-consistent relation extraction in this work. We discussed several language-consistent relation extractors that work in the closed domain. However, to the best of our knowledge, language-consistent open relation extractors are not yet proposed in current literature. Therefore, our fourth research question (How can we combine state-of-the-art open and multilingual relation extraction research to obtain a multilingual open relation extraction model?) is only answered within the cross-lingual domain in current literature. Since we belief that a language-consistent open model would be valuable for many web-based relation extraction applications, we propose a novel language-consistent open relation extraction model based on the design ideas behind the state-of-the-art closed language-consistent model (AMNRE) in the next chapter.

Additionally, we identified that aligning representations over languages can be highly beneficial for multilingual relation extractors. Multilingual word embeddings use information from multiple high-resource languages to create a shared embedding space in which also low-resource language can be represented. Since conventional multilingual word embedding techniques do not suffice to create embeddings for low-resource languages, unsupervised methods were introduced. These embeddings use only existing monolingual word embeddings to form aligned embeddings using an adversarial training approach. Both techniques provide a way to mimic the alignment behaviour over languages of AMNRE's adversarial training step on a word level, enabling usage in open relation extraction scenarios which are interpreted as sequence tagging problems.

6
LOREM

In this chapter, we continue to answer our fourth research question; How can we combine state-of-the-art open and multilingual relation extraction research to obtain a multilingual open relation extraction model? From Chapter 5, we concluded that this question is only partly answered by existing cross-lingual relation extraction literature. To tackle this challenge from the cross-lingual relation extraction domain, we present our Language-consistent Open Relation Extraction Model (LOREM). By combining two previously described models, we create a novel approach to extract open relations from multilingual corpora. To the best of our knowledge, this is the first neural open relation extractor that works in a language-consistent fashion.

6.1. Language-consistent open relation extraction model

In Chapter 5, we discussed cross-lingual and language-consistent relation extractors. Although cross-lingual relation extractors can provide a useful solution in situations where we want to use language pattern information from a target language to extract relations from a source language, they fail to exploit language-consistent patterns and require perfomant translators. Given the heterogeneity of languages used on the internet, we chose to turn our attention towards language-consistent relation extraction. We discussed that language-consistent relation extractors are currently not proposed within the open RE paradigm. Since we belief that such a model would be valuable for many web-based relation extraction applications, we propose a novel Language-consistent Open Relation Extraction Model; LOREM.

To construct this language-consistent open relation extraction model, we infuse the concept behind the state-of-the-art closed language-consistent model (AMNRE [81]) in the current state-of-the-art open relation extractor (NST [39]). More specifically, we train multiple NST models, one for each language (language-individual models) and one for all languages (language-consistent model). By combining these models, we end up with a set-up that is highly similar to AMNRE, yet for the open instead of the closed domain. We will describe LOREM in more detail using its visualization in Figure 6.1. As we can see in this visualization, an input sentence is fed into two NST models; a language-individual model that is specific to the language of the input sentence and a language-consistent model that is the same for all languages.

6.1.1. Input embeddings

An input sentence is encoded using two different pre-trained word embeddings. For the language-individual model, we use conventional pre-trained word embeddings. In Figure 6.1, these embeddings are represented in blue. The training sentences of the language-individual model all come from the same language. As a result, we expect that this model finds relation structures that are specific to that individual language.

The AMNRE model described in Section 5.1.2 uses an adversarial training method to align sentence embeddings from different languages. In order to achieve the same latent consistency among languages in LOREM, we use multilingual embeddings as explained in Section 5.3 for the language-consistent model. By using embeddings which are aligned between languages, we hypothesize that the burden of the CNN/BiLSTM layer to extract language-consistent patterns is eased. In Figure 6.1,

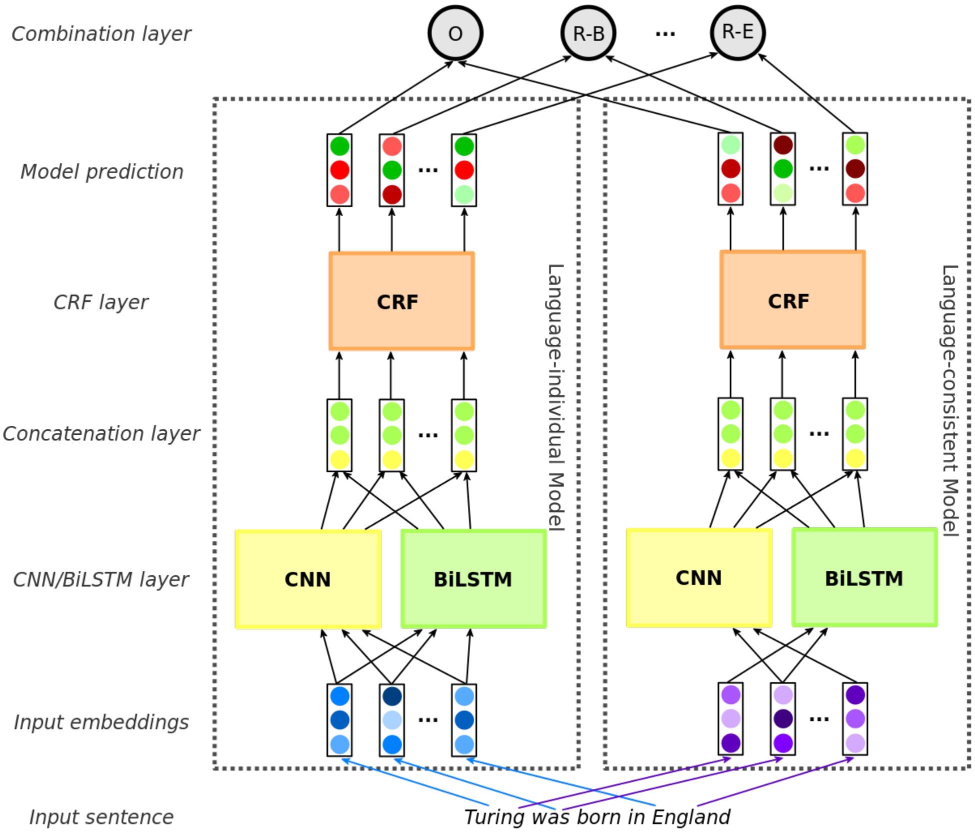


Figure 6.1: Architecture of our Language-consistent Open Relation Extraction Model (LOREM).

these embeddings are represented in purple. For this sub-model, the training sentences come from multiple languages. Since the model is jointly optimized on multiple languages, we expect this model to extract relation patterns consistent over these language.

In addition to word embeddings, entity embeddings are added to the input. These are simple one-hot encoded vectors which indicate if the current word is part of the first or second entity or not. Section 3.2.2 provides a more in-depth description of these embeddings. Please note that in contrast to the NST model, we do not use PoS embeddings since these introduce a dependency on PoS-taggers, which is not expedient for multilingual applications.

Mathematically, the input sentence is represented as a k-dimensional embedding sequence

$$\mathbf{x} = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_n\}, \tag{6.1}$$

where $\mathbf{w}_t$ is the representation of the t^{th} word of an input sentence that has n words. Here,

$$k = k_i + k_c, \tag{6.2}$$

k_i and k_c are the dimensonalities of the language-individual and -consistent model input respectively.

$$k_i = k_{mono} + k_e \text{ and } k_c = k_{multi} + k_e, \tag{6.3}$$

where k_{mono} is the dimensionality of the monolingual word embedding, k_{multi} of the multilingual word embedding and k_e of the entity tag vector.

6.1.2. NST layers

The next four layers are identical to the NST model described in Section 4.2.3. We shortly reiterate the steps in this model. A more detailed description can be found in the original NST paper [39] and in Section 4.2.3. During training, a separate language-individual model is trained for each language. By combining training sentences from multiple languages and using multilingual word embeddings, one language-consistent model is trained.

Relational words tend to occur in the neighbourhoods of entities. Therefore, certain parts of the input sentence might have a higher chance of containing relation words than others. A CNN is used to capture this local feature information from the input sentence. At the same time, a bidirectional LSTM is used to capture the forward and backward context of each word. The BiLSTM is especially well suited to capture long-range relations within the input sentence. By concatenating the outputs of the CNN and the forward and backward pass of the LSTM, a continuous representation of each word in the input sentence is formed. In Figure 6.1, the CNN output is represented in yellow and the BiLSTM output is represented in green. Next, these representations are used as the input for a straightforward CRF layer, which tags a word using the NST tagging scheme described in Section 4.2.3, which is reiterated in Table 6.1.

Tag	Meaning
R-S	Single word relation sub-string.
R-B	Beginning of relation sub-string.
R-I	Inside the relation sub-string.
R-E	Ending of relation sub-string.
O	Outside the relation sub-string.

Table 6.1: NST tagging as proposed by Jia et al. [39].

A CRF is used because it can efficiently consider correlations between past and future tags to predict the current tag. Therefore, CRFs are a common way to model sequence labelling tasks [39]. The output of the NST layers are two prediction sequences

$$\mathbf{y}_{ind} = \{\mathbf{i}_1, \mathbf{i}_2, ..., \mathbf{i}_n\} \text{ and } \mathbf{y}_{con} = \{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_n\}, \tag{6.4}$$

where $\mathbf{y}_{ind}$ contains the predictions of the language-individual model and $\mathbf{y}_{con}$ contains the predictions of the language-consistent model. $\mathbf{i}_t$ and $\mathbf{c}_t$ are the 5-dimensional prediction vectors of the language-individual and -consistent models respectively. For the original NST model, these are binary vectors which contain a 1 for the predicted tag and a 0 for all other tags. After our alteration, these vectors contain a probability score for each of the possible relation tags. This allows us to fittingly combine the predictions of the language-individual and -consistent models in the next layer.

Our model uses the hyper-parameters that were proposed by Jia et al. for their NST model [39]. The number of LSTM units is 200 and each convolutional filter has 200 feature maps. Parameters are optimized using the Adam optimizer [45]. The initial learning rate is 0.001 and it is reduced by a factor 0.1 if the loss function does not improve for some epochs. Moreover, dropout and early stopping are applied to regularize the model (see Section 3.2.4).

6.1.3. Combination layer

In the last layer, the predicted probabilities of the language-individual and -consistent models are combined. The combination method is the same as that of the AMNRE model. We define the final probability sequence by

$$\mathbf{y} = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}, \tag{6.5}$$

with

$$\mathbf{p}_t = \mathbf{i}_t \odot \mathbf{c}_t \tag{6.6}$$

for the t^{th} word in the input sentence[1]. The output tag sequence is defined by

$$\mathbf{z} = \{z_1, z_2, ..., z_n\}, \tag{6.7}$$

where

$$z_t = \arg\max_j \mathbf{p}_{tj} \tag{6.8}$$

and where $\mathbf{p}_{tj}$ is the j^{th} element of $\mathbf{p}_t$.

[1] $\odot$ is used as the Hadamard product.

Our model does not directly compute a confidence score for the extracted relation. There are however multiple ways in which we can derive a confidence score, by combining the probabilities for all individual words in the relation. Stanovsky et al. [75] experiment using different combination heuristics and find that multiplying the probabilities of the words that make up the relation provides the best fit for the ORE task. Please note that this approach favours short relations. Other approaches that could be used include taking the maximum, minimum or average probability of the individual words of the relation.

LOREM might yield tag sequences which are invalid. For example, the tag for a single word relation (*R-S*) can not be followed by a tag for the end of a multi-word relation (*R-E*). In this case, the first tag should be changed to *R-B* to form a valid tag sequence. We create two different versions of LOREM, LOREM$_{\text{clean}}$ which alters invalid sequences to valid sequences and LOREM which allows invalid sequences.

6.2. Conclusion

In conclusion, our described open relation extractor (LOREM) combines NST models inspired by the set-up of the closed language-consistent AMNRE relation extractor. We chose to use the NST model as the foundation of LOREM, since NST yields state-of-the-art results in the open relation extraction paradigm. If new and improved open relation extractors arise, they can easily replace the NST models in LOREM, given that the input and output of the sub-model remains unchanged.

We train language-individual models for separate languages by only using monolingual training sentences for each of these models. This enables these models to extract relation patterns that are specific to the individual languages. To train a language-consistent model, we create a training set by combining sentences from multiple languages. Following the concept presented by Wang et al. [81], this should enable the model to find language-consistent relation patterns. Given this observation, we hypothesize that the language-individual and -consistent models both extract different information. Therefore, combining both models should lead to a better overall performance on the open relation extraction task in multiple languages. In addition, we expect multilingual word embeddings to have a positive impact on the capabilities of the language-consistent model to extract relation patterns that are language-consistent. Since LOREM does not depend on external NLP tools or language-specific knowledge, it is relatively ease to include new languages. In this chapter, we propose LOREM as a potential answer to research question four; How can we combine state-of-the-art open and multilingual relation extraction research to obtain a multilingual open relation extraction model? Experiments conducted in Chapter 7 will need to show if the discussed hypotheses are indeed valid.

7 Experiments

In this chapter, we will describe experiments that are aimed at validating four main hypothesis which entail the expected behaviour of our Language-consistent Open Relation Extraction Model. We first describe these hypothesis, which are aimed at answering our fifth research question; How does the considered multilingual open relation extraction model hold on real-world test data? We then describe the used training and test data, after which we will focus on the experiments and their results.

7.1. Hypotheses

Wang et al. [82] show that including a language-consistent component results in performance gains within the closed high-resource relation extraction domain. For high-resource languages, we have large tagged open relation extraction datasets (100,000+ sentences) to our disposal. Since our model includes language-consistent information in a similar fashion for the ORE domain, the main hypothesis of this work is defined as;

H1: For *high*-resource languages, LOREM outperforms state-of-the-art monolingual open relation extractors by harvesting language-consistent relation patterns from multilingual texts.

In their work, Wang et al. [82] employ an adversarial training approach to achieve a sentence-level latent consistency among languages. We hypothesize that this can also be achieved on a word-level by using multilingual embeddings, resulting in hypothesis 2;

H2: Multilingual word embeddings improve the performance of the language-consistent sub-model, and thereby the performance of LOREM, by introducing a latent consistency among languages.

Next to our main hypothesis, we present initial experiments for no- and low-resource languages, based on the following definitions. For no-resource languages, we do not possess any tagged open relation extraction data. Hence, training a language-individual model for such a language is not possible. We only possess a limited set of training sentences (∼750 sentences) in the low-resource scenario. It differs from the high-resource scenario since this amount of training data will likely not suffice to train a performant language-individual model. The hypothesis are defined as follows;

H3: For *no*-resource languages, LOREM is able to outperform language-individual models by utilizing models of languages that have a similar origin.

H4: For *low*-resource languages, LOREM outperforms language-individual models by harvesting language-consistent relation patterns from multilingual texts.

7.2. Datasets

Information about the used training and test data is presented in Table 7.1. We selected these languages, since these are the only languages for which we could find openly available test data. The table shows that we used data from the following datasets:

WMORC [28] WMORC contains manually annotated ORE data for 3 languages (WMORC$_{\text{human}}$) and automatically tagged ORE data for 61 languages, bootstrapped using a cross-lingual projection approach (WMORC$_{\text{auto}}$). The sentences are gathered from Wikipedia.

NeuralOIE [20] Bootstrapped English dataset created by using only very high-confidence extractions of an existing relation extractor from Wikipedia sentences.

ClausIE [24] Three manually annotated English test sets from Wikipedia and New York Times sentences.

Raw Web/Parallel En-Sp [99, 100] Two manually annotated Spanish test sets from school text book and web page sentences.

	High					**No**	**Low**
	English	Spanish	French	Hindi	Russian	Italian	Dutch
# Training sentences	576,462	429,413	468,625	280,815	550,720	0	750
# Test sentences	2,191	246	512	622	573	10,000	100
Origin training data	NeuralOIE	WMORC$_{\text{auto}}$	WMORC$_{\text{auto}}$	WMORC$_{\text{auto}}$	WMORC$_{\text{auto}}$	-	WMORC$_{\text{auto}}$
Origin test data	ClausIE	RWP	WMORC$_{\text{human}}$	WMORC$_{\text{human}}$	WMORC$_{\text{human}}$	WMORC$_{\text{auto}}$	MC

Table 7.1: Description of the datasets used in our experiments for high-, no- and low-resource languages. Legend: *RWP* – Raw Web/Parallel En-Sp; *MC* – Manually Created

We randomly sample a subset of the available training sentences in order to keep the training time within a reasonable scope, the sizes of the sampled subsets are presented in Table 7.1. The size of our training sets is comparable to the dataset used in the original NST paper [39]. Moreover, early tests did not show substantial benefits of adding more data after this point. For Hindi we use the full set, since it is significantly smaller than the other datasets. We approach Dutch from a low-resource scenario, so we only sample 750 Dutch sentences. We don't use any Italian training data, since Italian is used as a no-resource language in our experiments. For training the language-consistent model, we again sample the datasets so that the combined set of all languages contains 450,000 - 550,000 training sentences. This way, we can make a fair comparison between the language-individual and -consistent sub-models since they are trained on the same amount of training sentences.

Given the very limited scope of existing multilingual open relation extraction literature, there are only very few results presented for these datasets. Moreover, these were the only publicly available ORE test sets we could find for non-English languages. For Dutch, we created our own test set by manually tagging 100 random Dutch Wikipedia sentences. Dutch was selected since this is the native language of the author of this work. The Italian test set is created by sampling 10,000 sentences from WMORC$_{\text{auto}}$. Since these sentences are automatically tagged, we do expect a higher noise level than in the manually tagged test sets.

For the language-individual model, we use FastText word embeddings [33]. These pre-trained embeddings are available for 157 languages. The embeddings were created using a CBOW model on a web crawl and Wikipedia dataset. The dimensionality of these vectors is 300. For the language-consistent model, we use pre-trained multilingual embeddings which are released by FastText [42] for 44 languages. The vectors were trained on a Wikipedia dataset and the dimensionality is also 300.

7.3. Comparison methods

During our experiments, we compare LOREM to a range of previously proposed methods, which were discussed in our literature review. We will shortly reiterate these systems. For English, we compare LOREM to the same baseline systems that were used during the evaluation of the NST model by Jia et al. [39]. These include:

NST$_{\text{no-PoS}}$ [39] The NST model forms the underlying model of LOREM, yet there are differences between the two. Most notably, the original NST model does not contain a language-consistent part. We present the results for NST without PoS-tags for a fair comparison.

Reverb [27] Reverb exploits syntactic and lexical constraints on binary relations expressed by verbs.

OLLIE [56] This model designs complex patterns using syntactic processing (e.g. dependency parsers).

ClausIE [24] ClausIE exploits linguistic knowledge about English grammar to detect and identify clauses and their grammatical function.

Open IE-4.x [55] This is a combination of a rule-based Open IE system and a system which analyzes the hierarchical structure between semantic frames to construct multi-verb open relation phrases.

For Spanish, we compare LOREM to the only two works which present results on the Spanish test set:

ExtrHech [100] A system based on syntactic constraints over PoS-tag sequences targeted at Spanish.

ArgOE [31] ArgOE uses dependency parsers to extract a set of propositions for different argument structures.

Finally, we compare LOREM to a **cross-lingual system** [28] presented by Faruqui et al.

7.4. Experimental results

7.4.1. Hypothesis 1: LOREM for high-resource languages

To validate our first hypothesis, we compare LOREM to several English open relation extractors. These are the same baseline systems that were used during the evaluation of the original NST model by Jia et al. [39]. Please recall that the NST model forms the underlying model of LOREM, yet there are significant differences between the two. First and foremost, the original NST model does not contain a language-consistent part. Accordingly, comparing it to LOREM can give us interesting insights in the validity of hypothesis 1 for the English language. Also, the original NST model relies on PoS-tags. We abundantly discussed the downsides of such dependencies in previous chapters. Since Jia et al. also presented results for NST without PoS-tags, we compare LOREM to this model for a fair comparison. Lastly LOREM is trained using a different dataset that is similar in size, since the NST training set was not published.

Table 7.2 contains the English test results. We see that both LOREM models outperform all baseline systems in terms of recall and F_1-scores. Focussing on the comparison with the NST model, we find that LOREM outperforms NST on precision, recall and therefore F_1-score. This indicates that adding a language-consistent component indeed improves the open relation extraction performance for the English language. The high F_1-scores of our LOREM models are mainly due to the excellent recall scores, compared to other systems. This means that LOREM is able to extract a large fraction of the relations that are present in the test set, without finding more invalid relations than the baseline systems (with the exception of OLLIE, which has an exceptional precision score at the cost of a very low recall score). We see that cleaning our predictions even further increases the recall, at the cost of precision. LOREM achieves the best presented F_1-score on the ClausIE datasets when PoS-tags are not used.

Model	**P**	**R**	**F$_1$**
LOREM	0.801	0.757	**0.782**
LOREM$_{\text{clean}}$	0.782	**0.765**	0.774
NST$_{\text{no-PoS}}$	0.783	0.708	0.744
Reverb	0.641	0.162	0.259
OLLIE	**0.985**	0.242	0.389
ClausIE	0.801	0.531	0.638
Open IE-4.x	0.792	0.331	0.467

Table 7.2: The average prediction results on the English datasets. The bolds indicate the best values.

In Table 7.3, we compare LOREM to two Spanish open relation extractors that are presented in the literature. It is important to note that both models heavily rely on semantic constraints and external NLP tools such as PoS-taggers. Even so much so that Gamallo et al. [31] state that most errors that are made by ArgOE are caused by the syntactic parser and PoS-tagger. They conclude that improving their system relies on the performance of other NLP tasks. For ArgOE [31] the authors only present a precision score.

The results show that LOREM is outperformed by ExtrHech on the Spanish datasets. It does however achieve a higher precision than the other multilingual model. Even though the evaluation results are not quite as high as the current state-of-the-art model, LOREM does have the big advantage that a user does not have to manually define semantic constrains. Relation structures are automatically extracted during the training process. From these results, we can however not confirm hypothesis 1 for the Spanish language.

Model	P	R	F_1
LOREM	0.615	0.522	0.564
$\text{LOREM}_{\text{clean}}$	0.585	0.547	0.564
ExtrHech	**0.710**	**0.595**	**0.647**
ArgOE	0.500	-	-

Table 7.3: The average prediction results on the Spanish datasets. The bolds indicate the best values.

We now turn our attention towards the three test languages that are included in the $\text{WMORC}_{\text{human}}$ dataset. To the best of our knowledge, there exists only one system for which results are published on these test sets, being the cross-lingual system by Faruqui et al. [28]. In their work, the writers only state precision scores. These can be found in Table 7.4. The source code for this model is not publicly released, which is why we can not compute the recall and F_1-scores ourselves.

We find that the cross-lingual model slightly outperforms LOREM in terms of the French precision score. However, LOREM clearly outperforms the cross-lingual model on both Hindi and Russian. This might be caused by the fact that the cross-lingual model is heavily dependent on a translator from English to the target language and an existing English relation extractor. It could be the case that this translator is better suited for French to English translations. Additionally, the relation structures in English and French might be more alike compared to the other languages, improving the suitability of the English relation extractor. LOREM eliminates this dependency by introducing a language-consistent component. The results indicate that this improves the generalizing capabilities over languages, providing prove for the validity of hypothesis 1.

Model	French			Hindi			Russian		
	P	R	F_1	P	R	F_1	P	R	F_1
LOREM	0.783	0.715	**0.747**	**0.900**	0.598	**0.719**	**0.762**	0.719	**0.740**
$\text{LOREM}_{\text{clean}}$	0.726	**0.729**	0.727	0.687	**0.618**	0.651	0.709	**0.726**	0.718
Cross-lingual	**0.816**	-	-	0.649	-	-	0.635	-	-

Table 7.4: The prediction results on the $\text{WMORC}_{\text{human}}$ datasets. The bolds indicate the best values per language.

In the previous experiments we only presented the results of the full LOREM model. One might wonder if only using the language-individual or language-consistent sub-model might not improve the performance. Therefore, we present the results obtained on all mentioned test sets in Table 7.5. From these results, we can derive several conclusions. First of all, LOREM generally outperforms both the language-individual and -consistent model. This means that combining these models leads to better results than using them separately. This observation falls in line with the conclusions presented by Wang et al. [81] for the closed domain and indicates the validity of hypothesis 1. More specifically, we find that LOREM yields the highest precision for all of the five language, albeit by a small margin for some of these languages. Moreover, the recall and F_1-scores are higher for four out of five languages. For the Russian language, the language-individual model performs slightly better in terms of recall and F_1-score.

In addition to these findings, we also observe a returning pattern between LOREM and $\text{LOREM}_{\text{clean}}$. For all languages, LOREM achieves higher precision and F_1-scores, indicating a better overall performance. However, cleaning the prediction results does consistently improve the recall of the model. From this, we conclude that LOREM generally outperforms $\text{LOREM}_{\text{clean}}$, yet $\text{LOREM}_{\text{clean}}$ should be used when recall is crucial for the application domain.

Another, somewhat surprising, observation from Table 7.5 is the reasonably good performance of the language-consistent model. Even though it is outperformed by LOREM and the language-individual model on all languages, the performance remains satisfactory given the fact that this model is not trained on one specific language. From these results, we wondered if relations structures truly differ a lot between languages. It could be the case that a language-individual model already performs

Model	**English**			**Spanish**			**French**			**Hindi**			**Russian**		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
LOREM	**0.801**	0.757	**0.782**	**0.615**	0.522	**0.564**	**0.783**	0.715	**0.747**	**0.900**	0.598	**0.719**	**0.762**	0.719	0.740
$\text{LOREM}_{\text{clean}}$	0.782	**0.765**	0.774	0.585	**0.547**	**0.564**	0.726	**0.729**	0.727	0.687	**0.618**	0.651	0.709	0.726	0.718
Language-individual	0.796	0.747	0.771	0.595	0.498	0.541	0.781	0.693	0.735	0.878	0.540	0.667	0.755	**0.741**	**0.748**
Language-consistent	0.792	0.734	0.762	0.583	0.471	0.521	0.733	0.673	0.702	0.813	0.566	0.667	0.712	0.690	0.701

Table 7.5: The prediction results of LOREM and its sub-models. The bolds indicate the best values per language.

reasonably well on other languages, eliminating the need for a language-consistent model. To test this hypothesis, we compare the average results of the language-consistent model over all five languages to the average results of the language-individual models on the languages. These results are presented in Table 7.6. The results clearly counteract the hypothesis, showing the merit of a language-consistent model over simply using one language-individual model for every language.

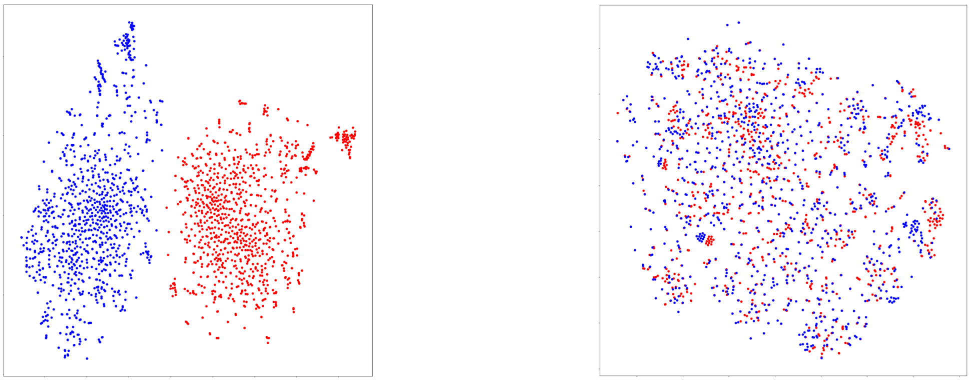
Model	**P**	**R**	**F_1**
Language-consistent	**0.727**	**0.627**	**0.671**
English language-individual	0.393	0.317	0.347
Spanish language-individual	0.586	0.390	0.455
French language-individual	0.679	0.464	0.543
Hindi language-individual	0.266	0.110	0.138
Russian language-individual	0.632	0.483	0.546

Table 7.6: The average prediction results of the language-consistent model and language-individual models on all test languages. The bolds indicate the best values.

7.4.2. Hypothesis 2: Multilingual vs. monolingual embeddings

The main contribution of our work is that we exploit language-consistent relation patterns to improve open relation extraction. Current multilingual relation extraction literature plainly utilizes monolingual word embeddings to encode sentences of different languages. However, when we use these embeddings as the input for our language-consistent model, we expect the model to extract patterns that are consistent over languages. Therefore, the model should ignore the language in which an input word is written. Naturally, aligning these word embeddings as discussed in Section 5.3 would ease the burden of the language-consistent model to extract language-consistent relation patterns.

This can be explained by using a visual example. In Figure 7.1, we have visualized 1,000 words in French and Spanish. We used t-SNE [54] to reduce the dimensionality of the embeddings from 300 to 2 for visualization purposes. In Figure 7.1a, we see that the words from both languages are easily separable. This makes it very hard for the model to find patterns that exist over several languages. In Figure 7.1b, we visualized multilingual embeddings after they were aligned. As we can clearly observe, the word embeddings are well mixed in this figure and language-consistent patterns are easier to find. This implies that we indeed obtain a high level of latent consistency using this technique. Please also note the similarity of these figures to Figures 5.2b and 5.2c, were the adversarial training step of AMNRE was visualized.

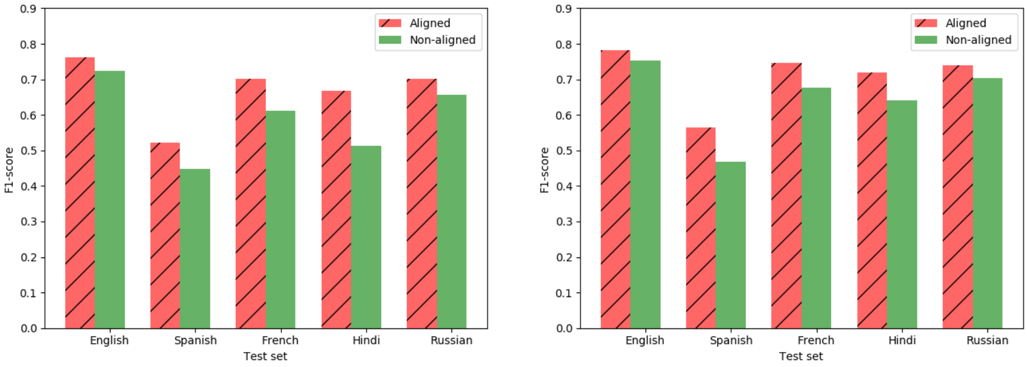


(a) French (red) and Spanish (blue) word embeddings *before* alignment.

(b) French (red) and Spanish (blue) word embeddings *after* alignment.

Figure 7.1: The visualization of non-aligned and aligned word embeddings using t-SNE.

To see if this approach indeed has an effect on the capacity of the language-consistent model to extract language-consistent patterns, we compare the results obtained by using both non-aligned (monolingual) and aligned (multilingual) word embeddings. In Figure 7.2, we present the results of this experiment on all five test languages. Additionally, we provide the impact of both approaches on the full LOREM model. From this figure, we can clearly observe that the aligned word embeddings yield better performance on every language for both the language-consistent sub-model and the full LOREM model. If we combine this observation with Figure 5.3, we can confirm the validity of hypothesis 2. Multilingual word embeddings can indeed be used to achieve the same latent consistency among languages in LOREM as is achieved by the adversarial training step of AMNRE. This improves the performance of the language-consistent model, and thereby of LOREM in general.



(a) F_1-score using aligned and non-aligned embeddings for the language-consistent sub-model.

(b) F_1-score using aligned and non-aligned embeddings for LOREM.

Figure 7.2: Aligned and non-aligned word embeddings for the language-consistent model and LOREM.

7.4.3. Hypothesis 3: LOREM for no-resource languages

If no open relation extraction training data is available for a certain language, our model can still be utilized in three possible ways: 1) we can use the language-consistent sub-model trained on other languages, 2) we can use a language-individual model of a language that has a similar origin to the current language 3) or we can combine both into a full LOREM model. In this experiment, we examine all three options on the Dutch and Italian test set. As is stated in the hypothesis, we do not use any open relation extraction training data. We do employ pre-trained Dutch and Italian word embeddings, taken from the FastText multilingual embeddings [42] which were also used in the previous experiments.

Model	**Dutch**			**Italian**		
	P	R	F_1	P	R	F_1
Language-consistent	0.705	**0.633**	0.667	0.506	**0.342**	**0.408**
English language-individual	0.655	0.582	0.616	0.293	0.232	0.259
Spanish language-individual	0.441	0.306	0.361	0.435	0.203	0.277
French language-individual	0.685	0.510	0.585	0.352	0.217	0.268
Hindi language-individual	0.000	0.000	0.000	0.362	0.029	0.054
Russian language-individual	0.703	0.265	0.385	0.393	0.164	0.232
LOREM	**0.744**	0.622	**0.678**	**0.554**	0.246	0.341
$\text{LOREM}_{\text{clean}}$	0.663	0.622	0.642	0.383	0.287	0.328

Table 7.7: The prediction results of the language-consistent model, language-individual models and LOREM on the Dutch and Italian test sets. The bolds indicate the best values per language.

Let us first focus on the Dutch test set. We start by using a language-consistent model that was trained on five languages, being English, French, Spanish, Russian and Hindi. This is the same language-consistent model that was used in previous experiments. By running this model on the Dutch dataset, we yield the results presented in Table 7.7. If we compare the results of the language-consistent model on the Dutch test set to the average result presented in Table 7.6, we find that they are highly similar. This implicates that the language-consistent model generalizes to other languages than the languages that it is trained on.

Table 7.7 also holds the results for the five language-individual models on the Dutch test set. We hypothesize that language-individual models of languages that have a similar origin as Dutch will yield better results than those of languages with a different origin. We define origin similarity using a language origin tree such as the one depicted in Figure 7.3. As an example, these trees allow us to quickly derive that an English model should perform better on the Dutch test set than a French model, according to the hypothesis.

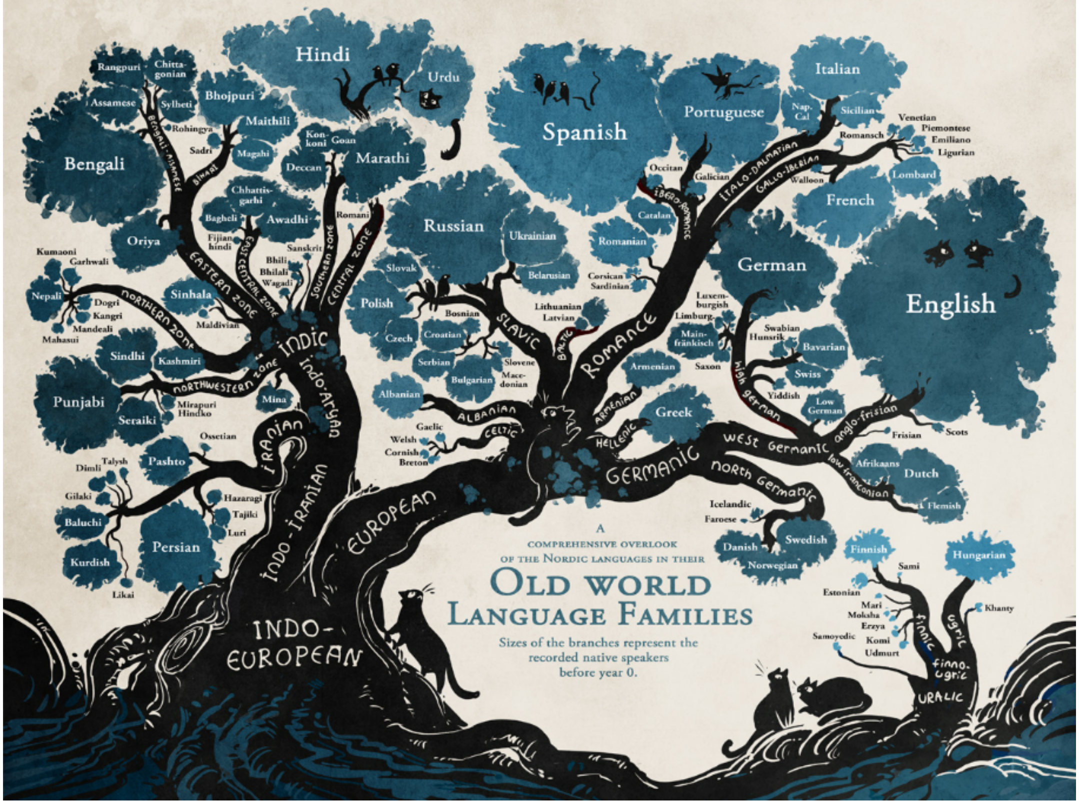


Figure 7.3: Language origin tree (source: `https://images.mentalfloss.com/sites/default/files/196.jpg`).

If we focus on the F_1-scores presented in Table 7.7, we find a general pattern that adheres to our intuition of utilizing languages that have a similar origin. The English model yields the highest F_1-score. This is to be expected since English and Dutch are the only two West Germanic languages in this experiment. The French model also performs reasonably well, this can be explained by the fact that French and Dutch are both of European origin. Albeit that French is a Romance and Dutch is a Germanic language. The only unexpected result is yielded by the Spanish model. Given that French and Spanish are both Romance languages, we would expect similar results on the Dutch test set. Yet, the Spanish model performs significantly worse than the French model. Looking at Table 7.6, we find that this behaviour also persists in the other languages that we experiment on. A possible explanation for this phenomenon could be the quality of the training sets, although both training sets are gathered from the same source. The Russian model also yields worse results than the French and English models. This is explained by the fact that Russian has a Slavic origin, which differs from the Dutch Germanic nature. Lastly, we find that the Hindi model is not able to find any valid relations. Given that all other language have a European nature and Hindi has an Indo-Iranian nature, this behaviour falls in line with our intuition.

Finally, Table 7.7 shows the results yielded by LOREM. To obtain these results, we combine the language-consistent model and the best performing language-individual model on the Dutch test set (English). These results fall in line with the conclusions derived from Table 7.5. LOREM again outperforms both sub-models, albeit by a small margin. In contrast to earlier findings on other languages, LOREM$_{\text{clean}}$ does not improve the recall over LOREM, while still yielding lower precision scores. Hence, LOREM is the better option in this scenario. In general, the best F_1-score obtained for the Dutch test set is slightly worse than the average F_1-scores of the languages presented in Table 7.5 (0.678 to 0.710). However, given the total absence of any Dutch training data we do argue that the results are satisfactory and provide an indication for the validity of our third hypothesis.

Table 7.7 also shows the results obtained on the Italian test set. Generally, we see that the models achieve lower scores compared to the Dutch test set. This could be caused by the fact that the models generalize better to some languages than to others. However, it could also be the case that this is caused by the noise that is present within the Italian set since it is not humanly annotated. That being said, we can still examine our intuition that languages of similar origin provide more useful information by comparing the evaluation scores obtained on the Italian test set. Indeed, we find a similar pattern within these results as we found in the Dutch test set, albeit less clear. Italian is a Romance language, as are French and Spanish. This explains why the French and Spanish models are the best performing language-individual models. Yet, the English model performs only slightly worse. A possible explanation is that the Italian test set contains English words that are frequently used in Italian sentences. The other two language do follow our intuition. Russian yields worse results, which is explained by the fact that it is a Slavic language instead of a Romance language. Hindi again yields a very low F_1-score, being the only non-European language.

Similar to the Dutch results, the language-consistent model outperforms all language-individual models. This shows the merit of combining languages to find language-consistent relation patterns. In contrast to earlier findings, the F_1-score is not improved by combining the language-individual and -consistent model in LOREM for the Italian test set. Although the precision score is improved, this comes at the cost of a lower recall score. Similar to previously discussed results LOREM$_{\text{clean}}$ obtains a higher recall score at the cost of the precision score, resulting in a lower F_1-score.

Even though the highest F_1-score is significantly lower than the average F_1-score on high-resource languages (0.408 to 0.710), we have to keep in mind that no Italian training data was used. Still we were able to extract many valid relations.

From the results presented in Table 7.7, we conclude that in general languages that are closely related to the test language provide better relation pattern information for the open relation extraction task. Yet, we only present results of a limited set of language-individual models on two test languages. Hence more experiments are needed to derive strong conclusions on the matter. We do hypothesize that training multiple language-consistent models for different groups of languages, instead of training a single language-consistent model, might further improve the performance on high-resource languages as well. This remains a topic for future work. It is also worth noting that these experiments show the first application of an open relation extractor on a different language than it was trained on without the need for a translator.

7.4.4. Hypothesis 4: LOREM for low-resource languages

If we only possess a very small training set of around 750 sentences, we again have multiple options to employ LOREM or its sub-models. Naturally, we could simply disregard the training data and approach the language as a no-resource language as was discussed in the previous section. However, unlike in the no-resource scenario, we are able to train a language-individual model on the small dataset.

Model	**P**	**R**	**F$_1$**
Language-individual	0.786	0.444	0.568
LOREM	0.753	**0.646**	**0.696**
Language-individual$_{\text{CNN}}$	0.727	0.404	0.519
LOREM$_{\text{CNN}}$	**0.836**	0.566	0.675
Language-individual$_{\text{LSTM}}$	0.716	0.535	0.613
LOREM$_{\text{LSTM}}$	0.802	0.616	**0.696**

Table 7.8: The prediction results of low-resource models on the Dutch test set. The bolds indicate the best values.

As an initial approach, we train a full language-individual model using the data that is available. To evaluate the model, we again use our hand-tagged Dutch test set. For training, we randomly select 750 Dutch sentences from the WMORC$_{\text{auto}}$ dataset [28]. The evaluation results for this model are shown in the first two rows of Table 7.8. If we compare the results shown in the top entry of this table to the evaluation results presented in Table 7.7, we find that the low-resource Dutch language-individual model is outperformed by the language-consistent, English language-individual and LOREM model. This indicates that a high-resource model in a different language outperforms a low-resource model in the test language, implying that we should not use a low-resource model and face the task as if it was a no-resource scenario. Yet, since we now have a Dutch language-individual model, we can combine it with the language-consistent model to form a full LOREM model. The results of LOREM are also presented in Table 7.8. If we again compare these results to Table 7.7, we see that the LOREM model that employs the Dutch language-individual model outperforms all models from the no-resource scenario. So even if the language-individual sub-model yields worse results, it can still contribute crucial information to the full LOREM model, enabling it to outperform high-resource models of similar languages. We conclude that LOREM outperforms the low-resource language-individual model for the Dutch test set, providing an indication of the validity of hypothesis 4.

Similar to the no-resource scenario, more experiments will be needed to derive solid conclusions on the matter. As new open relation extraction datasets arise, the evaluation of hypothesis 4 can be extended to increase the confidence in the conclusions for multiple languages.

Until now, we trained a full language-individual model for the low-resource language, ignoring the fact that we might need to treat a low-resource scenario differently than a high-resource scenario. As was depicted in Figure 6.1, a language-individual model contains both a CNN and an LSTM. Within the field of machine learning, it is a well-known phenomenon that more complex models generally require more training data, since more parameters need to be optimized. If we analyse the full language-individual model, we find that a total of approximately 1,000,000 parameters are optimized. Of these parameters, around 800,000 belong to the LSTM and 200,000 to the CNN[1]. Results presented by Jia et al. [39] clearly show that combining a CNN and LSTM outperforms both separate models for the high-resource open relation extraction task. This is also confirmed by initial tests that we performed. However, given the fact that we now only have a limited set of training samples, it might make more sense to only use either the CNN or LSTM for the low-resource language-individual model.

In Table 7.8, we present the evaluation results of the language-individual models that employ either a CNN or LSTM. Additionally, we show the performance of the LOREM models that use these language-individual models instead of the full language-individual model. We first examine the language-individual models. Here we see that the LSTM-based model outperforms the full language-individual model in the low-resource scenario. This could be explained by the fact that less parameters need to be optimized. On the other hand, the CNN-based model is clearly outperformed by both the LSTM-based and full language-individual model. This indicates that the CNN does not suffice to capture most relation patterns that are present in the texts. If we now turn our attention towards the LOREM models, we see that although LOREM$_{\text{LSTM}}$ achieves a higher precision than LOREM, this comes at the expense of a lower recall score. As a result, the F_1-scores of both models are exactly the same. Therefore, we did not find a clear advantage of simplifying the model in this low-resource scenario.

7.4.5. Qualitative analysis

Next to the quantitative analysis provided in the experiments above, we also carry out manual, qualitative analysis on English test sets [24] in order to gain a better understanding of the strengths and weaknesses of our model. We find that LOREM is able to extract the correct relations, even if multiple relations are present in a sentence. As a typical example, the sentence *“At least 8 schoolchildren were killed and at least 15 people were wounded when a deranged man burst into an elementary school and began stabbing students and teachers.”* contains multiple relations between different entities. If we provide the entity tuple `<a deranged man, students and teachers>`, LOREM extracts the relation `began stabbing`. However, if we select the tuple `<a deranged man, an elementary school>`, LOREM extracts `burst into`. Examples like these indicate that our model can deal with long, complex sentences.

In addition, we find that LOREM is better at extracting relations that follow abnormal patterns than the language-individual sub-model. For example, given the sentence *“The market wants to do better, said Gregory Bundy, head of equity trading.”* and entity tuple `<Gregory Bundy, The market wants to do better>`, the language-individual model does not find a relation, while LOREM extracts `said` as being the relation. Here, we find that the language-consistent component provides additional information which allows relations to be extracted, even if the entities appear in reverse order. It is likely that LOREM learned these relation patterns since they occurred in different training sets than the

[1] The other model layers only optimize around 4,000 parameters.

English set. Such examples illustrate the benefits of LOREM over a language-individual approach.

Lastly, we analyse common mistakes that are made by LOREM. Upon manual inspection, we find that the majority of errors arise from relations that contain multiple words. In these cases, LOREM extracts either too many or too few words compared to the ground truth relations. Typical examples include *“BIC is being sued by people who say their lighters exploded.”* and *“The region is still far from rebuilt.”*, from which LOREM extracts `is being sued` and `is still`, while the ground truth values are `is being sued by` and `is` respectively. These examples show that although the extraction is not completely correct, the relation is still captured reasonably well in many cases. Of course, the test set also contains sentences from which LOREM can not extract any relations. A typical error occurs when we want to extract relations that occur between more than two entities. Given a sentence like *“28 Square miles of antennae and computers that message smart fridges, robot lawn mowers and smart doorbells vacuum up satellite and radio communications.”* with entity tuple `<28 Square miles of antennae, radio communications>`, LOREM finds no relations even though the relation `vacuum up` is present between multiple entities in this sentence. The discussed examples indicate were improvements can still be made.

7.5. Conclusion

In this chapter, we presented a variety of different experimental results that entail the behaviour of LOREM and its sub-models. We specifically focussed on four main hypotheses, which provide an answer to our fifth research question; How does the considered multilingual open relation extraction model hold on real-world test data? During our experiments, we used English, French, Spanish, Russian, Hindi, Italian and Dutch data. We have shown that LOREM achieves state-of-the-art, or close to state-of-the-art, performance on most of these languages. From this, we conclude that monolingual open relation extractors can be improved by adding a language-consistent component. Moreover, typical examples from our manual analysis were presented, which showed the strengths and weaknesses of our model. Additionally, we showed that multilingual word embeddings can prove to be a crucial component of language-consistent relation extractors. This is the first work in which multilingual word embeddings are employed for the multilingual relation extraction task (either open or closed).

Next, we provided results which indicate that LOREM can be used, even if no open relation extraction training data is available for a language. By employing models that are trained on other, related languages, we were able to extract a significant number of valid relations from the test sets. We also showed that language-individual models of languages with a similar origin generally perform better than language-individual models of languages with a different origin. Although the language-consistent sub-model does not perform as well as the full LOREM model, it still yields satisfactory results on most languages, even on languages that it was not trained on. This shows that there indeed exist relation patterns that are consistent over languages.

If only a small training set is available, LOREM also proved to be a useful model. By combining an under-performing, low-resource language-individual model with a language-consistent model that is trained on high-resource languages, we were able to improve the performance. Experiments that were focussed on simplifying the model for low-resource scenarios did not lead to concrete improvements.

While reading these conclusions, one should keep in mind the number of languages used in our experiments. Although we use more languages in our experiments than in most of the experiments presented in related literature, more research will be needed to validate these conclusions. This especially holds for hypothesis three and four, for which we merely provide indications of their validity. Moreover, automatically created training sets were used for our experiments, which are expected to contain a certain level of noise. As new ORE training sets arise, more experiments can take place. Given the fact that our model does not rely on language-specific knowledge or external NLP tools, our intuition is that LOREM is applicable to many more languages than those discussed in this work. Future experiments will be needed to validate this intuition.

8 Conclusions and Discussion

In this work, we have presented the first Language-consistent Open Relation Extraction Model; LOREM. After defining the task description, we provided an extensive literature overview of the closed, open and multilingual relation extraction domains. From existing literature, we identified opportunities were combinations and improvements could be made. By infusing the idea behind the state-of-the-art multilingual closed relation extractor into the state-of-the-art open relation extractor, we were able to achieve state-of-the-art, or close to state-of-the-art, performance on most of our test languages. Our experiments show that harvesting language-consistent relation patterns improves language-individual open relation extractors. These results were obtained without relying on any language-specific knowledge or external NLP tools. As a result, it is relatively easy to extend LOREM to new languages. An open relation extraction training set and word embeddings are the only requirements to include a new language. Even when we do not possess such a training set, or only a very small training set, for a new language, our experiments indicate that LOREM and its sub-models can be used to extract valid relations, by utilizing information from related languages. Additionally, we conclude that multilingual word embeddings provide an effective approach to introduce latent consistency among input languages, which proved to be beneficial to the performance of our language-consistent model.

For real-world applications, we have to be aware of the practical implications of our results. By comparing our work to existing literature, it is tempting to get lost in minor improvements of evaluation metrics, which might not have a significant impact to the actual end user of the relation extraction system. The end user should keep in mind that the system is not perfect. On average, 77.2% of relations that are extracted by LOREM on our high-resource test languages are valid. Yet, the standard deviation of these precision scores is quite high, at 10.3%. This means that LOREM will work significantly better on some language than on others. On the recall side, our experiments indicate that LOREM finds 66.2% of the relations that are present in the texts. Again, the standard deviation is quite high at 9.8%. As was already described by Bronzi et al. [13], such a recall score is likely to be an optimistic approximation of the real-world recall of an open relation extraction system. This is due to the fact that there could be more relations present in the test set than those provided in the ground-truth file. That being said, the experiments do show that a large portion of the relations that are present are actually extracted for our test languages. Additionally, more experiments on different languages will have to be conducted to derive solid conclusions on the generalizing capabilities of the model over all languages, especially in no- and low-resource scenarios.

Next to these practical notes, we also identified that we currently know little about the specific reasoning of deep learning models for the relation extraction task. Although we have a general understanding of the contributions made by the CNN and RNN part of our model, the specific relation patterns that are extracted remain unknown. In multiple other domains where deep learning models are utilized, works have been proposed to obtain a better understanding of the reasoning behind these models by for example visualizing model layers [91]. Comparable efforts could result in a better understanding of the models, which could then lead to significant improvements within the neural RE domain.

Since the presented work can be used for many different applications, we also need to take into account the ethical footprint of such a system. To the best of our knowledge, there currently exist no literature which examines the ethical perspective on relation extraction systems, either open or closed. Yet, one could imagine that such a system could extract relations from enormous text corpora which would be close to infeasible to find by hand. If such a system were to be applied to for example create graphs between people from social media texts for marketing purposes, privacy comes into play and one should wonder if this use is desirable. Given our limited expertise on the field of ethics, we believe collaborations with the ethics community could shine a better light on this matter.

We believe that our work makes useful contributions to the field of multilingual open relation extraction. Naturally, our system is not perfect and many interesting directions of future work remain to be explored. Moreover, the presented conclusions hold on the datasets that are used in this work. Compared to existing literature, we evaluate our model on a bigger set of languages. Still, experiments on more datasets will be needed to increase the confidence in the generalizing capabilities of the model. Including LOREM in a full information extraction pipeline could also shine more light on the performance of the model when it is used in real-world applications. In conclusion, we are confident that our research is a step in the right direction towards solving the challenging multilingual open relation extraction task and we see many opportunities for future research within this promising research domain.

9 Future Work

In previous chapters, we have shown how LOREM incorporates ideas from multiple domains to form the first language-consistent open relation extraction model. Nonetheless, there are still many improvements that can be made, inspired by literature from the general open and closed relation extraction domains. In this chapter we will present our suggestions for future work on the matter.

9.1. CNN and RNN improvements

As was depicted in Figure 6.1, Convolutional and Recurrent Neural Networks form the basis of LOREM. In Section 3.2 we discussed a wide variety of improvements to these networks that were proposed within the closed relation extraction paradigm. Some of these improvements, such as Dropout and early stopping, are already applied in LOREM. Yet, we expect that more improvements could be made to the CNN and RNN by including more techniques proposed in the closed domain. These include the following:

- An in-depth analysis of the different layers of these models could shine a better light on which relation patterns are actually learned by the model. These insights could lead to significant improvements of the model. Additionally, one could argue that gaining a better understanding of the model's behaviour is an improvement on its own.

- Replace the global max-pooling layer in the CNN by a piecewise max-pooling approach to obtain a sentence-level representation that is more suited to the relation extraction task, as was proposed by Zeng et al. [94] in the closed domain.

- Use varying window sizes in the CNN to capture more close-distance relation patterns, as was proposed by Nguyen et al. [61] in the closed domain.

- Initialize the model weights by those of a model trained on a similar task. This technique is known as Transfer Learning [62]. We could for example initialize the model weights by that of the pre-trained model on a different language. In low-resource language scenarios, it could prove to be helpful to initialize the language-individual model weights by those of a similar high-resource language. Transfer learning could also be applied by pre-training the model on the similar named entity tagging task.

- We used the best performing hyper-parameters presented by Jia et al. [39], e.g. the number of LSTM units, the number of convolutional feature maps, the learning rate and so on. A more extensive investigation of these hyper-parameters could result in improvements.

9.2. Input improvements

The input that we use for our model is threefold; monolingual word embeddings, multilingual word embeddings and entity embeddings. Potential improvements can be obtained by experimenting with the following alterations:

- Instead of using straightforward entity markers, one could use position embeddings as was described in Section 3.2.2. In that section, we found that position embeddings do have a negative impact on the efficiency, but could improve the performance of the CNN.

- In this work, we only utilized pre-trained word embeddings from one source, being FastText [33, 42]. An extensive range of pre-trained word embeddings are freely available on the internet. By experimenting with these different embeddings, it could be found that another pre-trained embedding set dominates the FastText embeddings.

- In Section 5.3, we discussed unsupervised multilingual word embedding techniques that could be used to create multilingual word embeddings without the need for bilingual dictionaries or bilingual text corpora. This could prove to be helpful for low-resource languages. For the languages used in our experiments, multilingual embeddings were publicly released by FastText [42]. Yet this source only contains multilingual embeddings for 44 languages. Therefore, future work could focus on including unsupervised multilingual word embeddings to also facilitate languages that are not included by FastText.

9.3. General model improvements

Next to improvements to the CNN/RNN layer and the input, we also identify improvements and extensions that could be made to the full LOREM model. We believe that the following points could be interesting for future work:

- We currently use a multiplication approach to combine the language-individual and -consistent sub-models. However, many more possible combination approaches could be examined. For example, we could include a weighted average of the prediction probabilities. This does introduce extra parameters that need to be optimized, being the weights that should be accounted to the predictions of both sub-models. Initial tests using a weighted average with the same weights for both sub-models did not yield any improvements. Furthermore, we could argue that the prediction probabilities of both sub-models could form the input for a whole new classifier, which would naturally increase the complexity of the total model.

- Wang et al. [81] include an orthogonality constraint in their model to better separate the language-individual and -consistent feature spaces. A similar constraint could be included in LOREM for the same purpose.

- LOREM currently trains one single language-consistent model for all languages. Given the results presented in Section 7.4.3, it could be interesting to investigate the use of multiple language-consistent models for groups of languages with a similar origin.

- LOREM currently focusses on binary relations, being a relation between two entities. It could be interesting to see if the model could be generalized to deal with n-ary relations, as was discussed in Section 2.2.

- As was discussed in Section 4.3, multiple post-processing steps could be added to the model. By disambiguating and combining extracted relations and visualizing them in entailment graphs, the usability of the model output for real-world applications could be significantly increased.

- The NST foundation of LOREM could be replaced by a different neural relation extractor if better models arise, given that the input and output of the model remains unchanged.

9.4. Evaluation extensions

Now that we have presented our views on interesting future research on LOREM, we present extensions that can be made on our evaluation efforts:

- The WMORC$_{auto}$ that we use to train LOREM (except for English) is bootstrapped by an existing cross-lingual relation extractor. Therefore, we expect a certain level of noise in this dataset. The performance could be increased by creating training sets with lower noise levels, by for example using a consensus of multiple existing open relation extractors.

- Naturally, including more test languages during evaluation will lead to a better understanding of the performance of LOREM over languages. New test sets are needed to extend evaluation to other languages.

- During our experiments, we focussed on the validity of the model, not on the efficiency. Our workstation uses two NVIDIA P-1000 GPUs. Using these GPUs, our average training time for a language-individual or -consistent model is approximately 60 hours. Testing can be done in a matter of minutes, depending on the test set size. We do however not know how our system compares to previous systems in terms of efficiency and improvements to the efficiency can most likely be made.

- We evaluated LOREM using the evaluation procedures that are applied in existing open relation extraction literature. However, in Section 4.4, we described a different evaluation approach by Bronzi et al. [13]. Applying this evaluation method to our model could lead to more insights on the performance of LOREM on real-world data.

- Given the general set-up of LOREM as a sequence tagging model, we wonder if the model could also be applied to similar language sequence tagging tasks, such as named entity recognition. Exploring the applicability of LOREM to correlated tasks could be an interesting direction for future work.

Bibliography

[1] Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM Conference on Digital libraries*, pages 85–94. ACM, 2000.

[2] Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A. Smith. Massively Multilingual Word Embeddings. *CoRR*, abs/1602.01925, 2016.

[3] Waleed Ammar, Matthew Peters, Chandra Bhagavatula, and Russell Power. The AI2 system at SemEval-2017 Task 10 (ScienceIE): semi-supervised end-to-end entity and relation extraction. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 592–596, 2017.

[4] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 1, pages 344–354, 2015.

[5] Isabelle Augenstein, Diana Maynard, and Fabio Ciravegna. Distantly supervised web relation extraction for knowledge base population. *Semantic Web*, 7(4):335–349, 2016.

[6] Nguyen Bach and Sameer Badaskar. A review of relation extraction. *Literature review for Language and Statistics II*, 2, 2007.

[7] 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.

[8] Akim Bassa, Mark Kröll, and Roman Kern. GerIE-An Open Information Extraction System for the German Language. *J.UCS*, 24(1):2–24, 2018.

[9] David S. Batista, Bruno Martins, and Mário J. Silva. Semi-supervised bootstrapping of relationship extractors with distributional semantics. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 499–504, 2015.

[10] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155, 2003.

[11] Robert Blumberg and Shaku Atre. The problem with unstructured data. *Dm Review*, 13(42-49): 62, 2003.

[12] Sergey Brin. Extracting Patterns and Relations from the World Wide Web. In Paolo Atzeni, Alberto Mendelzon, and Giansalvatore Mecca, editors, *The World Wide Web and Databases*, pages 172–183, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg. ISBN 978-3-540-48909-2.

[13] Mirko Bronzi, Zhaochen Guo, Filipe Mesquita, Denilson Barbosa, and Paolo Merialdo. Automatic evaluation of relation extraction systems on large-scale. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 19–24. Association for Computational Linguistics, 2012.

[14] Peter F. Brown, Peter V. Desouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.

[15] Marc-Andr Carbonneau, Veronika Cheplygina, Eric Granger, and Ghyslain Gagnon. Multiple instance learning. *Pattern Recognition*, 77(C):329–353, 2018. ISSN 0031-3203.

[16] Xilun Chen and Claire Cardie. Unsupervised Multilingual Word Embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 261–270. Association for Computational Linguistics, 2018.

[17] Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. A Walk-based Model on Entity Graphs for Relation Extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 81–88, 2018.

[18] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[19] Jim Cowie and Wendy Lehnert. Information extraction. *Communications of the ACM*, 39(1): 80–91, 1996.

[20] Lei Cui, Furu Wei, and Ming Zhou. Neural Open Information Extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 407–413. Association for Computational Linguistics, 2018.

[21] Aron Culotta and Jeffrey Sorensen. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 423. Association for Computational Linguistics, 2004.

[22] Aron Culotta, Andrew McCallum, and Jonathan Betz. Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 296–303. Association for Computational Linguistics, 2006.

[23] Emanuele Damiano, Aniello Minutolo, and Massimo Esposito. Open Information Extraction for Italian Sentences. In *2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pages 668–673. IEEE, 2018.

[24] 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.

[25] Cicero dos Santos, Bing Xiang, and Bowen Zhou. Classifying Relations by Ranking with Convolutional Neural Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 1, pages 626–634. Association for Computational Linguistics, 2015. doi: 10.3115/v1/P15-1061.

[26] Hady Elsahar, Elena Demidova, Simon Gottschalk, Christophe Gravier, and Frederique Laforest. Unsupervised Open Relation Extraction. In *European Semantic Web Conference*, pages 12–16. Springer, 2017.

[27] 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*, EMNLP '11, pages 1535–1545, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4.

[28] Manaal Faruqui and Shankar Kumar. Multilingual Open Relation Extraction Using Cross-lingual Projection. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1351–1356. Association for Computational Linguistics, 2015. doi: 10.3115/v1/N15-1151.

[29] John R. Firth. A synopsis of linguistic theory, 1930-1955. *Studies in Linguistic Analysis*, 1957.

[30] Lisheng Fu, Thien Huu Nguyen, Bonan Min, and Ralph Grishman. Domain Adaptation for Relation Extraction with Domain Adversarial Neural Network. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, volume 2, pages 425–429. Asian Federation of Natural Language Processing, 2017.

[31] Pablo Gamallo and Marcos Garcia. Multilingual open information extraction. In *Portuguese Conference on Artificial Intelligence*, pages 711–722. Springer, 2015.

[32] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. *CoRR*, abs/1412.6572, 2014.

[33] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning Word Vectors for 157 Languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[34] Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 427–434. Association for Computational Linguistics, 2005.

[35] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational linguistics*, volume 2, pages 539–545. Association for Computational Linguistics, 1992.

[36] 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.

[37] Martin Hilbert and Priscila López. The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*, 2011. ISSN 0036-8075. doi: 10.1126/science.1200970.

[38] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

[39] Shengbin Jia, Yang Xiang, and Xiaojun Chen. Supervised Neural Models Revitalize the Open Relation Extraction. *CoRR*, abs/1809.09408, 2018.

[40] Jing Jiang and ChengXiang Zhai. A systematic exploration of the feature space for relation extraction. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 113–120, 2007.

[41] Xiaotian Jiang, Quan Wang, Peng Li, and Bin Wang. Relation extraction with multi-instance multi-label convolutional neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1471–1480, 2016.

[42] Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2984, 2018.

[43] Daniel Jurafsky and James H. Martin. *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2009. ISBN 0131873210.

[44] Nanda Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*, page 22. Association for Computational Linguistics, 2004.

[45] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980, 2014.

[46] Natalia Konstantinova. Review of Relation Extraction Methods: What Is New Out There? *Communications in Computer and Information Science*, 436:15–28, 2014. doi: 10.1007/978-3-319-12580-0_2.

[47] Michal Laclavík, Štefan Dlugolinskỳ, Martin Šeleng, Marcel Kvassay, Emil Gatial, Zoltán Balogh, and Ladislav Hluchỳ. Email analysis and information extraction for enterprise benefit. *Computing and Informatics*, 30(1):57–87, 2012.

[48] Omer Levy, Ido Dagan, and Jacob Goldberger. Focused entailment graphs for open IE propositions. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 87–97, 2014.

[49] Zhuang Li, Lizhen Qu, Qiongkai Xu, and Mark Johnson. Unsupervised Pre-training With Seq2Seq Reconstruction Loss for Deep Relation Extraction Models. In *Proceedings of the Australasian Language Technology Association Workshop 2016*, pages 54–64, 2016.

[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, pages 2124–2133, 2016.

[51] Yankai Lin, Zhiyuan Liu, and Maosong Sun. Neural relation extraction with multi-lingual attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 34–43, 2017.

[52] Angli Liu, Stephen Soderland, Jonathan Bragg, Christopher H Lin, Xiao Ling, and Daniel S Weld. Effective crowd annotation for relation extraction. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 897–906, 2016.

[53] Tianyi Liu, Xinsong Zhang, Wanhao Zhou, and Weijia Jia. Neural Relation Extraction via Inner-Sentence Noise Reduction and Transfer Learning. *CoRR*, abs/1808.06738, 2018.

[54] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[55] Mausam. Open information extraction systems and downstream applications. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 4074–4077. AAAI Press, 2016.

[56] Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 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*, EMNLP-CoNLL '12, pages 523–534, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[57] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *Proceedings of Workshop at ICLR '13*, 2013.

[58] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[59] 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, pages 1003–1011. Association for Computational Linguistics, 2009.

[60] Makoto Miwa and Mohit Bansal. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1105–1116. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1105.

[61] Thien Huu Nguyen and Ralph Grishman. Relation Extraction: Perspective from Convolutional Neural Networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48. Association for Computational Linguistics, 2015. doi: 10.3115/v1/W15-1506.

[62] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[63] Alexandre Passos, Vineet Kumar, and Andrew McCallum. Lexicon Infused Phrase Embeddings for Named Entity Resolution. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 78–86. Association for Computational Linguistics, 2014. doi: 10.3115/v1/W14-1609.

[64] Pengda Qin, Weiran Xu, and Jun Guo. An Empirical Convolutional Neural Network Approach for Semantic Relation Classification. *Neurocomputing*, 190(C):1–9, May 2016. ISSN 0925-2312. doi: 10.1016/j.neucom.2015.12.091.

[65] Pengda Qin, Weiran XU, and William Yang Wang. DSGAN: Generative Adversarial Training for Distant Supervision Relation Extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 496–505. Association for Computational Linguistics, 2018.

[66] Lizhen Qu, Gabriela Ferraro, Liyuan Zhou, Weiwei Hou, Nathan Schneider, and Timothy Baldwin. Big Data Small Data, In Domain Out-of Domain, Known Word Unknown Word: The Impact of Word Representations on Sequence Labelling Tasks. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 83–93. Association for Computational Linguistics, 2015. doi: 10.18653/v1/K15-1009.

[67] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics, 2009.

[68] 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.

[69] Ellen Riloff and Rosie Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the National Conference on Artificial Intelligence*, pages 474–479. AAAI, 1999. ISBN 0262511061.

[70] Octavian Rusu, Ionela Halcu, Oana Grigoriu, Giorgian Neculoiu, Virginia Sandulescu, Mariana Marinescu, and Viorel Marinescu. Converting unstructured and semi-structured data into knowledge. In *Roedunet International Conference (RoEduNet), 2013 11th*, pages 1–4. IEEE, 2013.

[71] Leandro MP Sanches, Victor S Cardel, Larissa S Machado, Marlo Souza, and Lais N Salvador. Disambiguating Open IE: identifying semantic similarity in relation extraction by word embeddings. In *International Conference on Computational Processing of the Portuguese Language*, pages 93–103. Springer, 2018.

[72] Stephen Soderland, Brendan Roof, Bo Qin, Shi Xu, Mausam, and Oren Etzioni. Adapting Open Information Extraction to Domain-Specific Relations. *AI Magazine*, 31:93–102, 2010.

[73] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[74] Gabriel Stanovsky and Ido Dagan. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2300–2305, 2016.

[75] Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 885–895, 2018.

[76] 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.

[77] Laurent Vannini and Hervé Le Crosnier. *Net.lang: Towards the Multilingual Cyberspace*. C & F Editions, 2012. ISBN 9782915825244.

[78] Ngoc Thang Vu, Heike Adel, Pankaj Gupta, and Hinrich Schütze. "Combining Recurrent and Convolutional Neural Networks for Relation Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 534–539. Association for Computational Linguistics, 2016. doi: 10.18653/v1/N16-1065.

[79] Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. ACE 2005 Multilingual Training Corpus LDC2006T06. Philadelphia: Linguistic Data Consortium, 2006.

[80] Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. Relation Classification via Multi-Level Attention CNNs. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1298–1307. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1123.

[81] Xiaozhi Wang, Xu Han, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Adversarial Multi-lingual Neural Relation Extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1156–1166, 2018.

[82] Yanshan Wang, Liwei Wang, Majid Rastegar-Mojarad, Sungrim Moon, Feichen Shen, Naveed Afzal, Sijia Liu, Yuqun Zeng, Saeed Mehrabi, Sunghwan Sohn, et al. Clinical information extraction applications: a literature review. *Journal of Biomedical Informatics*, 77:34–49, 2018.

[83] Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. Universal decompositional semantics on universal dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723, 2016.

[84] Michael Wick, Khashayar Rohanimanesh, Aron Culotta, and Andrew McCallum. Samplerank: Learning preferences from atomic gradients. *Advances in Ranking*, page 69, 2009.

[85] Hong woo Chun, Yoshimasa Tsuruoka, Jin dong Kim, Rie Shiba, Naoki Nagata, and Teruyoshi Hishiki. Extraction of gene-disease relations from Medline using domain dictionaries and machine learning. In *Proc. PSB 2006*, pages 4–15, 2006.

[86] Fei Wu and Daniel S. Weld. Open information extraction using Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 118–127. Association for Computational Linguistics, 2010.

[87] Yi Wu, David Bamman, and Stuart Russell. Adversarial training for relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1778–1783, 2017.

[88] 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.

[89] 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, pages 956–966, 2014.

[90] Alexander Yates and Oren Etzioni. Unsupervised methods for determining object and relation synonyms on the web. *Journal of Artificial Intelligence Research*, 34:255–296, 2009.

[91] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[92] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3(Feb):1083–1106, 2003.

[93] 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.

[94] 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.

[95] Dongxu Zhang and Xiaoyang Tan. Relation Classification via Recurrent Neural Network. *CoRR*, abs/1508.01006, 2015.

[96] Sheng Zhang, Kevin Duh, and Benjamin Van Durme. MT/IE: Cross-lingual open information extraction with neural sequence-to-sequence models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, volume 2, pages 64–70, 2017.

[97] Sheng Zhang, Kevin Duh, and Benjamin Van Durme. Selective Decoding for Cross-lingual Open Information Extraction. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, volume 1, pages 832–842, 2017.

[98] Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. Bidirectional long short-term memory networks for relation classification. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pages 73–78, 2015.

[99] Alisa Zhila and Alexander Gelbukh. Comparison of open information extraction for English and Spanish. In *19th Annual International Conference Dialog*, pages 714–722, 2013.

[100] Alisa Zhila and Alexander Gelbukh. Open Information Extraction from real Internet texts in Spanish using constraints over part-of-speech sequences: Problems of the method, their causes, and ways for improvement. *Revista signos*, 49:119 – 142, 03 2016. ISSN 0718-0934.

Acronyms

Acronym	Meaning
ACE	Automatic Content Extraction
AFT	After the second entity
AMNRE	Adversarial Multilingual Neural Relation Extraction [81]
BEF	Before the first entity
BET	Between both entities
BiLSTM	Bidirectional Long Short-Term Memory
CBOW	Continuous Bag of Words [57]
CNN	Convolutional Neural Network
DANN	Domain Adversarial Neural Network [30]
DIPRE	Dual Iterative Pattern Relation Expansion [12]
DNN	Deep Neural Network
GAN	Generative Adversarial Network
HAC	Hierarchical Agglomerative Clustering
IE	Information Extraction
LOREM	Language-consistent Open Relation Extraction Model
LSTM	Long Short-Term Memory [38]
MIL	Multiple-Instance Learning
MIML	Multi-Instance Multi-Label
MLP	Multi-Layer Perceptron
MNRE	Multilingual Neural Relation Extraction [51]
MT/IE	Machine Translation/Information Extraction [96]
MUC	Message Understanding Conference
NER	Named Entity Recognition
NLP	Natural Language Processing
NST	Neural Sequence Tagging [39]
OIE	Open Information Extraction [7]
PCA	Principal Component Analysis
PCNN	Piecewise Convolutional Neural Network [94]
PMI	Pointwise Mutual Information
QA	Question Answering
RE	Relation Extraction
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
TF-IDF	Term Frequency-Inverse Document Frequency
t-SNE	t-Distributed Stochastic Neighbor Embedding [54]
UMWE	Unsupervised Multilingual Word Embedding [16]
WOE	Wikipedia-based Open Extractor [86]