TEXTUAL RELATION EXTRACTION WITH EDGE-ORIENTED GRAPH NEURAL MODELS

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF SCIENCE AND ENGINEERING

2020

Efstathia Christopoulou

Department of Computer Science

Contents

Word Count: 46,810

List of Tables

List of Figures

Abstract

TEXTUAL RELATION EXTRACTION WITH EDGE-ORIENTED GRAPH NEURAL MODELS

Efstathia Christopoulou

A thesis submitted to The University of Manchester
for the degree of Doctor of Philosophy, 2020

Textual Relation Extraction is an important task for Natural Language Processing that aims to detect semantic relations between named entities in text. It can be seen as a multi-aspect challenge, with varying applications to several downstream tasks such as Question Answering, Knowledge Base Completion and Event Extraction. In this dissertation, we aim to address two of the most common sub-tasks of Relation Extraction: the detection of relations inside sentences, also known as intra-sentence RE, as well as across sentences, known as inter-sentence RE.

Our objectives in this study are two fold. Firstly, we suggest that interactions between multiple pairs should be taken into account when modelling relations, so as to enrich pair representations. Secondly, we want to leverage information encoded in the connections between different pairs, rather than the entities of the pair alone. To realise both goals, we propose a novel graph-based neural model, which we call edge-oriented; that is, it exploits the edges of a graph, which by definition correspond to relations, in the form of multi-dimensional representations. The proposed model can construct and/or update edge representations between pairs of nodes using other edges in the graph. As a result, we simultaneously model multiple pairs in a textual snippet by forming their representations as multi-hop interactions between their arguments. Throughout this work, we validate the proposed approach on several datasets, showing that it effectively improves relation detection on both multi-pair and single-pair sentences in different domains.

Regarding document-level relations, we further propose a simple but intuitive way to construct heterogeneous document-level graphs and infer interactions between their

nodes. We suggest that simple graph structures that can be constructed with heuristics can effectively capture interactions of interest in documents. In addition, incorporating information from the entire document proves beneficial for both intra- and inter-sentence relations. Overall, our edge-oriented model achieves promising results, thus demonstrating its potential suitability for relation extraction and other graph-based tasks.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

It appears that this is the beginning of the end of a long journey which would have been impossible without the help of numerous people. I am thankful to everyone who encouraged me to keep going and inspired me to strive for something better all these years. Thank you to everyone from the NLP community who showed interest in my work.

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Sophia Ananiadou, for accepting me as her student, for the opportunities she offered me, the trust she placed in me and for her constant enthusiasm about my work. Her continual support makes me feel honoured to have had her as my supervisor.

I am additionally indebted to Prof. Makoto Miwa, who has been my Deep Learning and Relation Extraction mentor. The work of this PhD has been assisted by his guidance and resourceful discussions and it has been a privilege for me to be able to work alongside him.

I would also like to thank Prof. Jun'ichi Tsujii for giving me the opportunity to intern in AIST/AIRC for three months. I also extend my gratitude to Prof. Hiroya Takamura for introducing me to the KIRT team, and also to all of its members for the truly inspiring discussions and nice meals we shared during my internship in Tokyo.

An additional thank you goes to my examination committee, Prof. Ion Androutsopoulos and Dr. André Freitas for their invaluable suggestions to improve my thesis, as well as their interest in my work. Moreover, thank you to everyone I met in ACL in Australia, ACL in Italy and EMNLP in Hong Kong, something that made attending conferences one of the most exciting experiences. Thank you for all the interesting discussions and fun times.

A big thank you goes to all of my former and current colleagues in NaCTeM, who have supported me through thick and thin. I would have never been able to do this without you guys: Meizhi, Chryssa, Maolin, Kurt, Thy, Paul, Nhung, Austin, Axel, Matt, Piotr, Sunil, Jock, Phong.

Another big thank you is due to all of my friends from Greece as, despite the distance, we still maintain the same undergrad-like friendship. Thank you for all the mental support and tolerance: Nikolas, Konstantina, Thodoris ($\times 3$), Dimitris, Foteini, Dorothea, Xristos, Ioanna, Alex, Marina, Filippos, Kostas, Elisavet, Petros, Leonidas.

Finally, this would not have been possible without the support of my extended family, their never-ending mental boost, endless love and their encouragement for whatever I choose to do. I really want to thank my sister, Dimitra, for being a good flatmate and for helping me proofread this thesis; some seriously fun times. Last, but not least, I want to thank Chris for being by my side through it all, and almost becoming a computer scientist himself because of it. Thank you for always making me believe in myself.

Abbreviations

ADE Adverse Drug Event

CNN Convolutional Neural Network

DDI Drug-Drug Interactions

DS Distant Supervision

EHR Electronic Health Record

GCN Graph Convolutional Network

GRU Gated Recurrent Unit

LSTM Long-Short Term Memory

MIL Multi-instance Learning

MLE Maximum Likelihood Estimation

MLP Multi-layer Perceptron

NEL Named Entity Linking

NIF Negative Instance Filtering

OIE Open Information Extraction

PPI Protein-Protein Interactions

RE Relation Extraction

RNN Recurrent Neural Network

SDP Shortest Dependency Path

TL Transfer Learning

TRE Textual Relation Extraction

Chapter 1

Introduction

1.1 Motivation

Humans identify the world around them by "relating" themselves to their surroundings. In other words, they form connections with the environment they interact with. As Aristotle states in Metaphysics "*Things are called "relative" (a) In the sense that 'the double' is relative to the half, and 'the triple' to the third; and in general the 'many times greater' to the 'many times smaller' [...]*"[a] (Ross, 1925). As such, the connections individuals form are but a part of a greater 'network', where everything has some degree of relation with everything else; nothing is irrelevant.

One major component of achieving those connections is by developing communicative mechanisms. While interactions with the environment primarily involve the senses (touch, sight, smell, etc.), humans, in particular, developed a unique way to communicate with each other, which occurred in the form of language. Throughout history, language evolved to accommodate human needs through both oral and written discourse. Written discourse, i.e. texts, further evolved to adapt to new concepts and notions and became gradually different according to the domain it addressed. Different genres emerged as a result, such as literary, medical or legal texts, among others. Such texts address particular audiences and feature jargon (i.e. specialised vocabulary) that is unique to each domain.

Every piece of discourse, whether oral or written, must follow a "parameter" in

[a] "πρός τι λέγεται τὰ μὲν ὡς διπλάσιον πρὸς ἥμισυ καὶ τριπλάσιον πρὸς τριτημόριον, καὶ ὅλως πολλαπλάσιον πρὸς πολλοστημόριον καὶ ὑπερέχον πρὸς ὑπερεχόμενον: [...]" [Aristotle, Metaphysics, 5.1020b]

order to facilitate communication; that is, coherence. Coherence, as a principle, ensures that both words and phrases, as well as notions and concepts, by extension, can adequately form meaning and, eventually, pass along a message to another person. The skill to first understand and then produce language is the most vital ingredient for human communication, written or oral.

Nowadays, there has been an increasing interest to comprehend this ability further. This research is of particular interest to the domain of Artificial Intelligence (AI) as, by understanding this complex human function, we will be able to construct models that automatically perform the same operation. The abundance of written text, particularly in digital form, additionally enhances the interest in how humans understand language, as well as how they understand and form new associations between lexical elements. Since it is not humanly possible to read through such large amounts of text, in order to extract important information more effectively, we seek automatic ways to trace facts, opinions or evidence of known or hidden relations between people, objects or other concept entities through text. Natural Language Processing is a field of AI that aims to develop methods to not only encode but also produce language to or from meaningful representations, for computers.

A part of this field involves developing a particular set of methods for the automatic identification of named entities in text, as well as the associations between them. The automatic identification of semantic associations (relations) between named entities is named *Relation Extraction* and is the main subject of this study.

In this dissertation, we particularly study *textual relation extraction*, by focusing on automatic ways to detect *semantic associations* between named entities in specific textual snippets.

1.1.1 Why Graphs?

The semantic associations between entities or concepts are defined by humans and as a result we can categorise relations in text to be either explicit or implicit. The former indicates the presence of adequate information in the textual snippet that allows us to identify the corresponding association. The latter indicates the absence of explicit contextual information, hence human inference is required to extract the association. For instance, in the example "*John is the son of Peter and Peter is the son of Robert*", there is clearly a relation between John and Robert. However, this is not explicitly stated in the sentence.

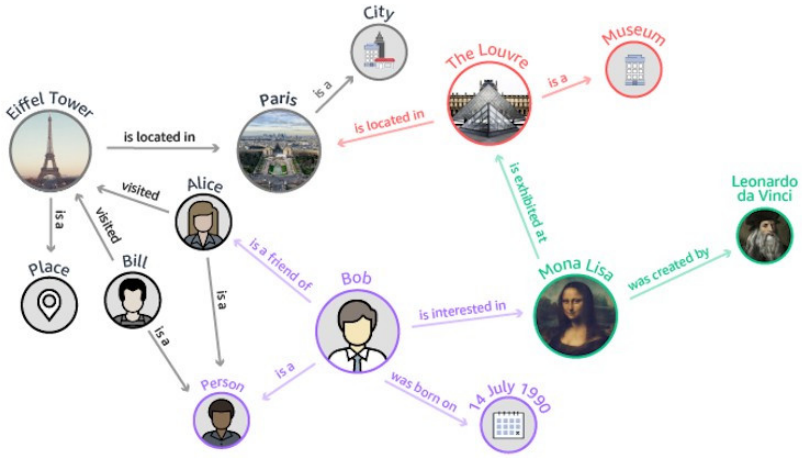


Figure 1.1: Example of a semantic graph. Source: towardsdatascience.com

In order to extract these implicit relations, inference is required, i.e. the formation of conclusions based on given facts or assumptions that are supposed to be true. Inference is prominent in written discourse. It typically requires the use of existing associations to form new ones; in other words, it involves the interaction of multiple elements in its entirety. A Semantic Network (Allen and Frisch, 1982) (also known as semantic graph) is one of the oldest forms of knowledge representation as a graph structure, connecting concepts (nodes) with semantic associations (links). Figure 1.1 shows an example of a semantic graph, which connects places, cities and people with semantic relations.

Graph structures provide a comprehensive visualisation of associations between different concepts, as well as a way to infer relationships simply by connecting the dots. For instance, in the example of Figure 1.1, since Alice visited the Eiffel Tower and the Eiffel Tower is located in Paris, we can infer that Alice visited Paris. In Natural Language Processing, and in written discourse in particular, the identification of semantic associations between entities is realised via the task of Relation Extraction (RE). Approaches for RE that utilise graph structures have attracted particular attention over the last years where, with the help of neural networks, they have achieved significantly high performance on existing datasets (Luan et al., 2019).

Graphs are the primary representation of relations, hence their importance in the construction of graph-based methods for automatic relation extraction. Existing graph-based methods used in various NLP tasks consider words as nodes and interactions among them as edges. However, they focus more on the nature and informativeness of

the relation participants (i.e. arguments or nodes in the graph) instead of the connections between them (edges in the graph). Recently, there have been several efforts to incorporate, or give more significance, to these connections (Gong and Cheng, 2019). We also aim towards this direction by proposing a method that focuses more in meaningfully representing graph edges with multi-dimensional features instead of graph nodes.

Our general goal is to address the problem of relation extraction from textual sources, with respect to a given piece of text, as a graph-based problem and in particular to effectively model multiple interactions among elements in text. This goal is twofold. Firstly, it involves constructing simple graph structures from the input text using named entities and secondly, includes the formation of meaningful representations between the interactions of these entities using the graph structure.

Our proposed methodology is *edge-oriented* in the sense that the relations between two nodes are formed by unique, multi-dimensional edge representations which are directly used to model interactions in a graph.

1.2 Research Questions, Hypotheses and Objectives

With regard to the problem we aim to study, we form the following research questions (RQ), accompanied by their respective hypotheses (H).

RQ_1 In cases where multiple entities exist in a sentence, could we take advantage of all intra-sentence entity-to-entity interactions to improve detection of semantic relations?

- H_1 The relation between two named entities in a sentence can be supported by the interactions of these entities with other, co-existing named entities in the same sentence, in a joint training setting.

RQ_2 Can multiple interactions among entities in a sentence be beneficial for detecting relations in other domains?

- H_2 Modelling multiple interactions among pairs in a sentence can be effective for relations in both the generic and the biomedical domain.

RQ_3 Can we model documents as heterogeneous graph structures and infer document-level relations?

$H_{3.1}$ We can map documents to partially connected, heterogeneous graph without the need for syntactic dependency structures.

$H_{3.2}$ Document-level inference, i.e. using information from the entire document, is beneficial for both intra- and inter-sentence relations.

Based on the research hypotheses proposed, we establish the following research objectives (O):

O_1 Develop a sentence-level, relation extraction model that is independent of external resources and syntactic tools, in order to directly incorporate knowledge from other pairs in the same sentence in a meaningful way.

O_2 Propose an edge-oriented graph encoding mechanism for relation extraction in sentences, which focuses on the meaning of the connections between entities, and study its potential effectiveness and suitability in relation extraction tasks.

O_3 Validate the proposed approach on sentential datasets belonging to different domains (news and bio-medicine), while investigating the variations and similarities observed in relations between different domains.

O_4 Propose an intuitive way to construct document-level graphs without using syntactic dependency tools, in an effort to simplify the task of identifying relations across sentences.

O_5 Investigate the effect of the proposed edge-oriented approach on a heterogeneous, document-level graph for detection of intra- and inter-sentence relations.

1.3 Contributions

This study makes the following contributions (C), associated with the previously presented objectives, as summarised below:

C_1 We propose a new methodology that models sequences as graph-based structures, particularly for relation extraction in sentences. The proposed approach utilises contextual information from all the pairs in the sentence and does not require external syntactic tools.

C_2 We develop a graph-based algorithm that models multi-dimensional edge representations instead of node representations. The algorithm iteratively composes multi-hop walks between two entities into edge representations.

C_3 We adapt the proposed approach to the biomedical domain, where we prove that interactions between multiple pairs of named entities can further improve the detection of associations between other entities in both domains.

C_4 We propose a simple methodology to model documents into graphs with heterogeneous types of nodes and edges, without the requirement for external syntactic tools. The graph is constructed using simple heuristics that stem from the natural associations between elements in a document.

C_5 We apply our edge-oriented algorithm on heterogeneous document-level graphs and improve the detection of relations inside and across sentences.

C_6 We test the developed models on both human and automatically annotated corpora showing the effectiveness of our proposed method even when named entities and relations are noisy.

1.3.1 Publications

A large amount of the work proposed in this dissertation has been already published. This dissertation contains existing, improved or additional results with relevance to the following publications, as discussed in the corresponding chapters.

- A Walk-based Model on Entity Graphs for Relation Extraction
  - ⋆ Association for Computational Linguistics (ACL)
  - ⋆ Christopoulou, Miwa, and Ananiadou (2018)

- Adverse Drug Events and Medication Relation Extraction in EHRs with Ensemble Deep Learning Methods[b]
  - ⋆ Journal of American Medical Informatics Association (JAMIA)
  - ⋆ Christopoulou, Tran, Sahu, Miwa, and Ananiadou (2020)

[b]This is a co-authored paper and the contributions are discussed in detail in Chapter 5.

- Inter-sentence Relation Extraction with Document-level Graph Neural Network[c]
  - ⋆ Association for Computational Linguistics (ACL)
  - ⋆ Sahu, Christopoulou, Miwa, and Ananiadou (2019)

- Connecting the Dots: Document-level Neural Relation Extraction with Edge-oriented Graphs
  - ⋆ Empirical Methods for Natural Language Processing (EMNLP-IJCNLP)
  - ⋆ Christopoulou, Miwa, and Ananiadou (2019)

1.4 Dissertation Structure

The dissertation consists of two introductory chapters, three main content chapters and the conclusions chapter.

In Chapter 2, we provide key definitions regarding the general task of Relation Extraction. In the first part of the chapter, we discuss the multiple aspects of the task, as well as the evaluation metrics that are typically used. In the second part, we investigate existing methods developed for relation extraction over the years by categorising them in a meaningful taxonomy. We further present structured-based methods in detail and discuss their advantages and disadvantages for relation extraction. Chapter 3 is a brief introduction to neural components, some of which are further used in the following chapters, describing their principal functionality.

Chapter 4 examines our initial hypothesis (H_1). We introduce an edge-oriented graph-based relation extraction model and explain the motivations behind the proposed approach. We describe in detail the model architecture and validate it on three textual, sentence-level datasets from the general domain (news articles or encyclopedias). Finally, extensive analysis is conducted on the different model components to better understand the behaviour of the model. In this chapter, we include parts of our published work Christopoulou et al. (2018).

We further adapt and validate our approach on the biomedical domain in Chapter 5. At this point, we address our second hypothesis (H_2) and attempt to prove the effectiveness of the proposed approach across different domains. Experiments are conducted with textual data from scientific articles and electronic health records. This chapter includes work from our published paper Christopoulou et al. (2020).

[c]The contents of this paper are not included in this dissertation.

In Chapter 6, we extend the proposed model to relations that reside across sentences and deal with document abstracts in particular. Our proposed model is evaluated on a different relation extraction setting, where the objective is to identify relations between named entities mapped to Knowledge Bases. We describe a simple method to construct graphs from documents and address our last two hypotheses ($H_{3.1}$ & $H_{3.2}$) by incorporating our edge-oriented mechanism on document-level graphs. We further discuss the suitability of this approach to the task at hand. In this chapter, we include work from our published paper Christopoulou et al. (2019).

In the final Chapter 7, we summarise the findings of each individual chapter and draw important conclusions from the overall study. Finally, we elaborate on limitations of the existing work as well as plans for future work.

Chapter 2

Relation Extraction: An Overview

In this chapter we aim to present a thorough overview of Relation Extraction, both from the perspective of the various tasks that can be associated with it, as well as in terms of developed methods to tackle these tasks. We first provide key definitions regarding important components involved in relation extraction tasks, such as entities and relations. We then split existing tasks into structured categories depending on the information one aims to extract. The remainder of the chapter discusses in detail existing approaches across various dimensions, elaborating on their advantages and limitations.

2.1 Definitions

Textual Snippet A textual snippet is defined as a small piece of a text. Phrases, sentences, paragraphs or documents constitute typical representations. The simple term *snippet* will be used henceforth to express textual snippets.

Word A *word* is a single distinct meaningful element of language. Words always have a meaning when found alone in discourse, contrary to *morphemes* that sometimes need another morpheme to form something meaningful. Essentially, morphemes are the building blocks of words.

Named Entity A *named entity* is a word or a group of words that constitute a proper name, such as a person, an object, a location, an organisation. Named entities are usually referred to as simply *entities*.

Entity Mention An occurrence of a named entity is called a named *entity mention*.

Entity Concept Named entities can occur multiple times in a snippet under the same name, a synonym or an alias (including abbreviations). *Entity concepts* correspond to an entity class where entity mentions can be mapped. In Figure 2.1 the named entities *Apple* and *Apple Computer* can be mapped to the entity concept *Apple Inc.* as multiple mentions of that concept.

In the following sections we will explicitly distinguish between *entity mentions* and *entity concepts*.

Entity Type Named entities typically belong to a particular semantic category. As a result, it is common to assign semantic *entity types* to them, e.g. *organisation*. Entity types are also known as *entity semantic categories*. Both terms will be used interchangeably.

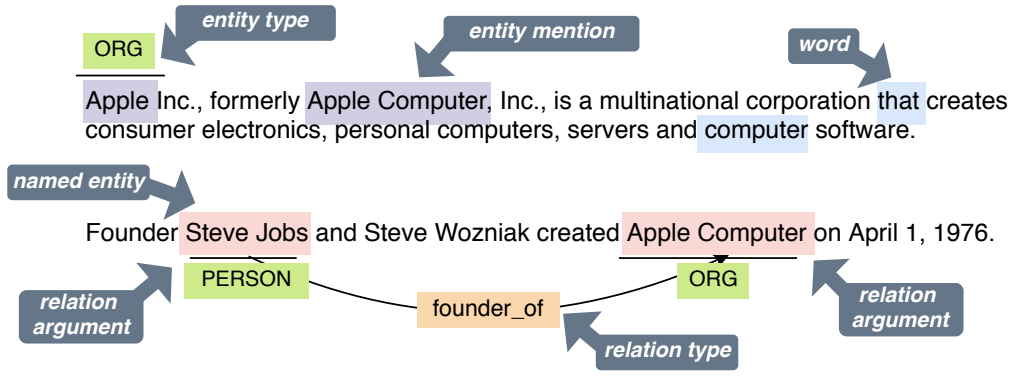


Figure 2.1: Example illustrating the different definitions for entities and relations.

Relation Type Similarly to entity types, the *semantic relation type*, *relation category* or simply *relation type* describes the semantic category of the relationship between two or more named entities.

Relation Instance A *relation instance* is a group of named entities participating in a relation along with their relation type. In case of two named entities, a relation instance can be defined as a triple (e_1, r, e_2), where e_1 and e_2 correspond to the named entities and r corresponds to their relation type. As shown in Figure 2.1 the triple (Steve Jobs, founder_of, Apple Computer) is a relation instance, or as commonly used, a *relation triple*.

Relation Argument In a relation instance, each of the named entities that participate in the relation are named *relation arguments*. In the following sections we will refer to relation arguments as *target entities* or *entities of interest*.

Pair A pair is essentially a relation instance that includes only two relation arguments. For simplicity, when referring to the relation between two named entities we will use the term *pair*.

2.2 Associated Tasks

Textual Relation Extraction (TRE) is an umbrella term that describes the identification of relations between elements in text. RE can be divided into multiple tasks that focus on particular types of relations, formulated in different settings. A taxonomy of related tasks is summarised in Figure 2.2.

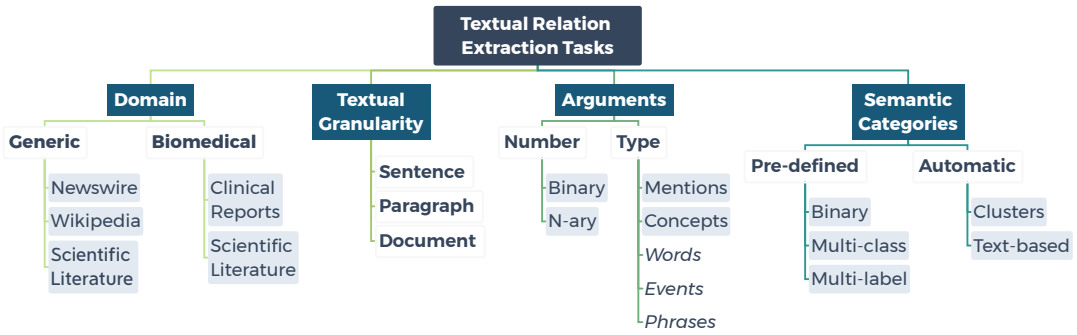


Figure 2.2: Categories of relation extraction tasks according to different aspects of extracted information.

An initial distinction can be made based on the target **domain**. Approaches for relation extraction were initially applied to generic data, extracted from the web or from encyclopedias such as Wikipedia. However, due to the increasing interest of natural language processing in extracting structured information for the biomedical domain, multiple methods have been adapted or developed particularly for the biomedical domain, including the scientific literature or clinical reports.

The next more general category refers to the **textual granularity** from which we aim to extract relations between named entities. Sentence-level RE, i.e. extraction of relations in sentences, is the most well-studied RE task. Recently, inter-sentence RE, i.e. extraction of relations beyond sentences, started to gain more interest from the scientific community (Gupta et al., 2019). In reality, most existing relations are inter-sentential. This setting is common in paragraph- or document-level snippets, that consist of multiple sentences.

We can then separate relation tasks with respect to the **number of arguments** participating in the relation. In the case of two entities being involved in a relation, the

task is defined as *binary* RE, which is the most commonly-studied RE setting. Apart from *binary* RE, recently there has been an increasing interest in *n*-ary RE, where the objective is to identify the relation between *n* number of named entities in a structured form (Song et al., 2018). The **type of arguments** between which we aim to extract a relation can also be considered a different RE task. In particular, *mention-level* RE indicates extraction between named entity mentions, while *concept-level* RE refers to extraction of relations between concepts. As defined earlier, concepts can be viewed as generalised named entities, typically mapped to unique Knowledge Base identifiers. Concept-level RE is more common in the biomedical domain, where entities can be expressed via multiple names. In this setting, *multi-instance learning* (Carbonneau et al., 2018) is usually incorporated in order to learn from the many instances or forms that a concept can appear in textŒ.

Another RE categorisation is defined based on the **semantic categories** that named entities can share. When relation extraction is treated as a classification problem, relations are typically chosen from a set of pre-defined relation categories. The simplest categorisation is binary classification. This setting involves only two possible relation categories, i.e. *related/non-related*. In the case of a larger set of relation categories, e.g. *located-in*, *founder-of*, etc., the classification is named *multi-class*. Typically, in multi-class classification, only one relation category is allowed for each pair. Finally, if a given group of entities can be assigned more than one potential relation type simultaneously, the classification is known as *multi-label*. However, when relation extraction is treated as an open information extraction setting, relation categories are extracted automatically from text, hence named *text-based*, or are latently induced as *clusters*.

In general, a combination of the aforementioned settings is often targeted by researchers, such as mention-level *n*-ary RE, concept-level multi-class document-level RE, and so on. There is also another group of tasks which focuses on more specific relations. In more detail, causal relation extraction (Blanco et al., 2008) aims to identify relationships of causality between given named entities. Such relations require identification of indicative words for causality or extraction of patterns between cause-effect relations. Temporal relation extraction (Ling and Weld, 2010) is another specific RE task that aims to detect relations of temporality between named entities or events, e.g. *before*, *after*, *simultaneously* etc. Again, linguistic properties of text such as tenses, or the order of the narrative should be considered to detect such relations. Finally, joint extraction of named entities and relations is a common approach, which is often referred to as *end-to-end named entity and relation extraction*. The objective is

to simultaneously extract both named entities and relations between them, either in a pipeline manner of through joint training. In contrast, simple RE assumes that named entities are already annotated in text (either by humans or other methods). Joint extraction has been proven to be an effective method, as relation detection training can assist named entity recognition (NER) (Miwa and Bansal, 2016; Bekoulis et al., 2018b).

Many downstream NLP tasks include RE as one of their main information extraction steps. Such an example is event extraction, a task that aims to identify a complex set of relationships between entities, that constitute an event (Hogenboom et al., 2011). Events typically require the identification of a trigger word (a key word that specifies the event), as well as the semantic relations of other words with the trigger. The task of identifying these relations is named *event argument role detection* and can be considered an intermediate step for event identification. Another task that falls in the same category is Knowledge Based Population (KBP) (Getman et al., 2018). In this task, the goal is to augment an existing Knowledge Base (KB) with additional information. An aspect of this task is called *slot-filling*, where the objective is to identify all possible information for a specific entity. This is highly correlated with relation extraction as all information about an entity can be identified through the relations of this entity to other elements. Other tasks that considerably benefit from RE are Question Answering (Ostapov, 2011), Fact Verification (Thorne and Vlachos, 2018), etc.

2.2.1 Challenges

The aforementioned tasks involve several other challenges besides the target problem they aim to solve. Relation Extraction, in general, is a complex procedure that, in an ideal scenario, involves tackling several linguistic phenomena. We only name a few here that are the most commonly encountered, but it should be noted that there is a large variety of such phenomena, due to the diversity of language.

Aliases are occurrences of words in different surface forms, that refer to the same thing. For example, *New York City* and *NYC* refer to the same city, but the former consists of three words, while the latter is an abbreviation. Named entity linking is the task used to map multiple such occurrences into a single concept; it is additionally useful for RE so as to extract more general associations. Co-reference is related to aliases, with the difference that the referents cannot only be nouns or names, but also pronouns. The identification of co-reference is often targeted in RE in order to extract associations that might exist in different sentences. Another phenomenon is polysemy, where each word can have multiple meanings, according to accompanying

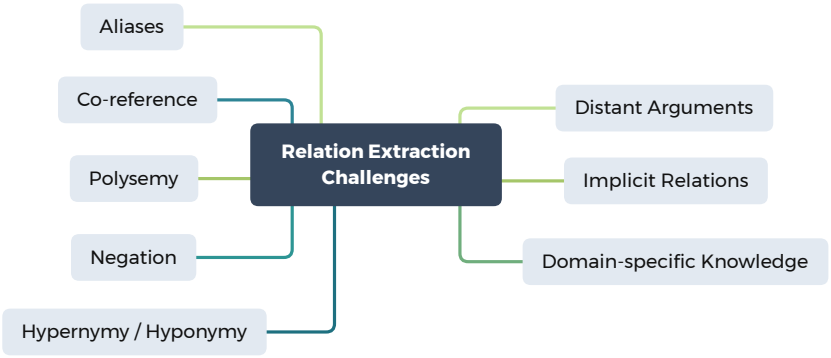


Figure 2.3: A few of the existing challenges in Relation Extraction challenges.

context. Despite existing models that do well in disambiguating entities, there are several challenging cases, as entities can still have multiple semantic types despite being in a single context. Hypernymy and hyponymy refer to the presence of hierarchies between entities, which are typically challenging when inferring relations. For instance, if there is a relation between animals and food, it might be that the same relation stands for dogs, cats etc. However, if there is a relation between cats and food, it does not necessarily mean that the relation can be generalised to all animals. Knowledge about the hierarchy of entities provides helpful insight for relation extraction. It is typical that ontologies are incorporated into RE models to solve such issues. Moreover, negation is a common grammatical phenomenon that directly affects relations by altering their meaning. Negation is not always easy to detect since it can be expressed in various ways. Nevertheless, it is something that should be generally taken into account in RE applications.

Three other challenges that perhaps are not purely linguistic, concern distant arguments, implicit relations and domain knowledge. Regarding the first, detection of long-distanced relations is a goal that has been targeted for a long time and continues to do so until today. The distance between arguments is often a negative factor for their association, i.e. typically, if two entities are too far apart in a snippet, the chance of being related is small. However, this is not always the case, as, when sentences are long, they might include several parenthetical phrases before mentioning the second participant, leading to longer distance. This issue is generally targeted from RE methods achieving promising results over the years. Implicit relations, on the other hand, are mostly studied in discourse, i.e. extraction of relations between clauses. However, they are also apparent in classic RE, when there is no explicit evidence to support the

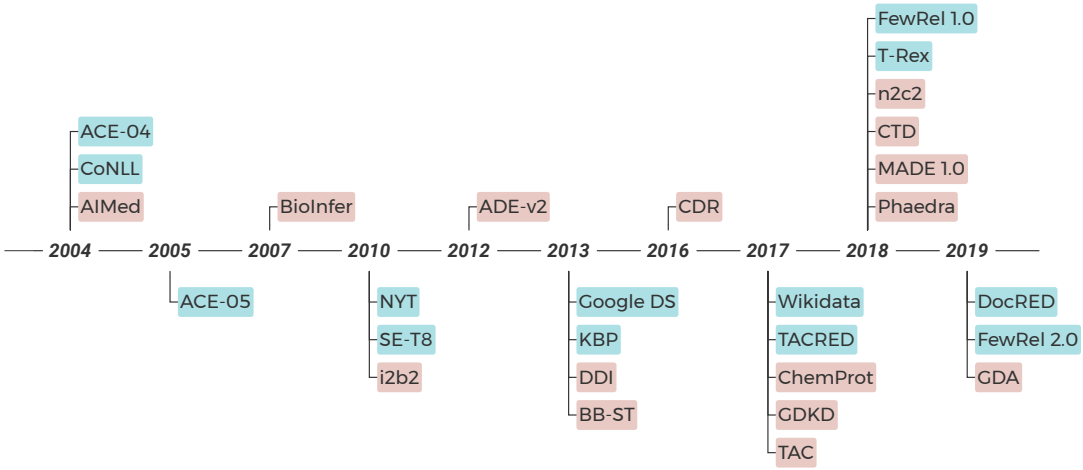


Figure 2.4: Timeline of Relation Extraction datasets. Red boxes correspond to datasets developed for the biomedical domain, while blue boxes correspond to datasets developed for the general (news) domain.

underlying relation; although it can be understood from the general context. Additionally, domain knowledge is another important factor in RE. Different corpora from different domains use different vocabularies, hence rendering the identification of relations difficult to adapt to other domains without external domain-specific knowledge. This type of information can be found in Knowledge Bases or domain-specific annotated corpora, which are often employed by RE methods to facilitate the detection of entities in domain-specific corpora.

2.2.2 Datasets and Corpora

Over the years, several datasets have been created to assist the development of RE models. The majority of these datasets have been annotated by humans for different domains, known as *gold* annotated corpora. Recently, RE corpora have been created automatically with the help of Distant Supervision (DS) (Chapter 2.3.4), also known as *silver* corpora. For the general domain, RE datasets typically contain multiple named entities of different semantic types and consequently multiple semantic relation categories among pairs. For the biomedical domain, existing datasets developed for RE mostly include only two types of named entities or concepts, as well as two relation types between them (i.e. related or non-related). This difference can be attributed to the fact that biomedical corpora are much more challenging to annotate as they require considerable domain expertise.

Figure 2.4 illustrates the timeline of creation of different relation extraction datasets

for the news-wire and the biomedical domains. It is worth noting that, in recent times, more and more datasets are being created either automatically or human annotated, since older datasets have reached high performances with recent advances in relation extraction methods. Another interesting observation is that the datasets developed for the biomedical domain seem to be more compared to the ones developed for the news-wire domain. This can be attributed to the fact that relation extraction is much more challenging for the biomedical domain, as biomedical relations are much more complicated, most of the time requiring domain expertise.

We summarise a list of existing and most commonly used relation extraction datasets for both domains in Table 2.1. The datasets are categorised based on their domain, textual granularity, classification, arguments and type of annotation. An official relation extraction task was firstly formulated at the seventh Message Understanding Conference (MUC-7) in 1998, introducing three semantic entity types (organisation, location, artifact) and three semantic relation categories (location-of, employee-of, product-of). Later on, the Automatic Content Extraction (ACE) project started by defining more semantic entity and relation categories both coarse and fine-grained. The first ACE domain datasets developed were developed in 2003, 2004 and 2005 (Doddington et al., 2004), containing named entities, relations and events in various languages, mostly from news articles, and have been extensively used by many existing relation extraction approaches. Another early introduced dataset was that of *CoNLL 04* (Roth and Yih, 2004), which also contains entities and relations for the general domain. However, it is much smaller than ACE in terms of number of sentences. A few years later, the *SemEval-2010 shared task 8* (Hendrickx et al., 2010) was held, where a general domain, sentence-level dataset was created for binary relation extraction. The dataset was widely used to improve relation classification, assuming gold named entities were given in advance.

The New York Times (NYT) corpus was created by Riedel et al. (2010) using distant supervision. The dataset was constructed by aligning Freebase[a] and the New York Times (NYT) corpus with data between 2005 and 2006 for training and data from 2007 for testing. The *KBP13* dataset was introduced by Angeli et al. (2014) for multi-instance multi-label (MIML) relation extraction. They utilised the 2010 and 2013 KBP document collections and a snapshot of Wikipedia 2013 as raw text corpus to perform distant supervision. In a similar manner, *WikiData* was created by Sorokin and Gurevych (2017) with Wikipedia relations using distant supervision. Google also

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

constructed their own distantly supervised corpus from Wikipedia[b]. One recent dataset is the *TACRED* dataset (Zhang et al., 2017c). It is a large scale relation extraction dataset with 23 different semantic entity and 41 semantic relation categories. It was build by querying the English TAC KBP newswire and web forum corpus with target entity mention pairs of interest. T-Rex (Elsahar et al., 2018) is a recently created dataset using distant supervision from Wikipedia. In addition, DocRED is the first generic domain dataset for document-level relations extraction (Yao et al., 2019). Finally, two additional datasets have been introduced for few-shot relation extraction, FewRel 1.0 (Han et al., 2018) and FewRel 2.0 (Gao et al., 2019). The latter contains annotations from both domains.

Moving on to the biomedical domain, Protein-Protein Interaction was the first relation extraction task that was targeted. The *AIMed* (Bunescu and Mooney, 2006) PPI dataset was constructed from 225 PubMed abstracts and annotated with binary interactions between human proteins. Similarly, the *BioInfer* dataset (Pyysalo et al., 2007) is a smaller PPI dataset. The *i2b2* Challenge (Uzuner et al., 2011) aimed at identifying relations between biomedical concepts from clinical records, specifically between three types of pairs; *treatment-medical problem*, *medical problem-test* and between *medical problems*, including 11 relation categories in total. The *ADE-v2* dataset was created by Gurulingappa et al. (2012a) and contained relations between Adverse Drug Events (ADEs) and drugs. The PHAEDRA corpus (Thompson et al., 2018) is a semantically annotated corpus for pharmacovigilance with annotations of entities, relations, events and coreference. Another very well studied dataset is the *DDI*, Drug-Drug Interactions dataset (Herrero-Zazo et al., 2013). The dataset contains annotated drugs and their associations from two different sources, MEDLINE and DrugBank. It was used as gold standard in the SemEval-2013 DDI Extraction Task (Segura-Bedmar et al., 2013). The *BioNLP 2013 Bacteria Biotopes Shared Task* (Bossy et al., 2013) introduced relations between bacteria and biotopes. The BioCreative V *CDR* dataset (Li et al., 2016a) contains document-level binary relations between chemical and disease concepts. It was the first gold dataset for inter-sentence relation extraction in biomedical abstracts of full documents. BioCreative VI *ChemProt* (Krallinger et al., 2017) targeted Chemical-Protein relations and is a large sentence-level relation extraction dataset with multiple fine-grained relations between chemicals and proteins. The *n2c2* dataset (Henry et al., 2019) considered relations between drugs and other medication-related entities, including 9 different relation categories. (Roberts et al., 2017)

[b] https://ai.googleblog.com/2013/04/50000-lessons-on-how-to-read-relation.html

Dataset	Domain	Gran/ty	Class/n	Arg. Type	Annot.
ACE04	news	sentence	MC	mention	gold
CoNLL	news	sentence	MC	mention	gold
ACE05	news	sentence	MC	mention	gold
NYT	news	sentence	MC/ML	concept	DS
SE-T8	news	sentence	MC	mention	gold
GoogleDS	news	sentence	MC	concept	DS
KBP	news	sentence	MC	mention	DS
WikiData	wikipedia	sentence	MC	mention	DS
TACRED	news	sentence	MC	mention	DS
FewRel 1.0	wikipedia	sentence	MC	mention	DS+gold
T-Rex	wikipedia	sentence	MC	mention	DS
DocRED	news	abstract	MC	mention	DS+gold
FewRel 2.0	wikipedia/ biomedical	sentence	MC	mention	DS+gold
AIMed	biomedical	sentence	binary	mention	gold
BioInfer	biomedical	sentence	binary	mention	gold
i2b2	clinical	sentence	MC	mention	gold
ADE-v2	biomedical	sentence	binary	mention	gold
DDI	biomedical	sentence	MC/binary	mention	gold
BB-ST	biomedical	sentence	binary	mention	gold
CDR	biomedical	abstract	binary	concept	gold
ChemProt	biomedical	sentence	MC	mention	gold
GDKD	biomedical	document	binary	mention	DS
TAC	biomedical	sentence	MC	mention	gold
n2c2	clinical	sentence	MC	mention	gold
CDT	biomedical	abstract	MC	concept	DS
MADE 1.0	clinical	sentences	MC	mention	gold
Phaedra	biomedical	sentence	MC	mention	gold
GDA	biomedical	abstract	binary	concept	DS

Table 2.1: Existing Relation Extraction Datasets. *Gran/ty* refers to Granularity, *Class/n* refers to Classification, *Arg.* refers to Argument, *Annot.* refers to Annotation, *MC* stands for Multi-class classification, *ML* stands for Multi-label classification and *DS* stands for Distant Supervision.

Concerning distantly supervised datasets, Verga et al. (2018) built a large corpus for abstract-level relation extraction named *CTD*, by aligning PubMed abstracts with the *CTD* database. Similarly, Wu et al. (2019) build *GDA* by aligning PubMed abstracts with the *DisGeNet* (Piñero et al., 2016) database. Quirk and Poon (2017) built a distantly supervised corpus for binary relation extraction between genes and drugs

using the Gene-Drug Knowledge Database (GDKD), while they later incorporated the *CIVIC*[c] database to extend the corpus for ternary relation extraction through drug-gene-mutation triples. Both datasets were constructed to extract relations from full-text documents.

2.2.3 Evaluation Metrics

The performance of relation extraction systems is typically measured with a set of evaluation metrics. The most commonly used metrics include counting the number of correct and incorrect predictions of a relation extraction model in comparison with some ground truth annotations. We report the most commonly used metrics designed to estimate the performance of RE models, as described below, considering that named entities are given.

Firstly, it is necessary to define errors/statistics that are used to compute these metrics. Let us assume that we have a relation extraction problem with only two relation categories: a pair shares a relation, or it does not share a relation. TP (*True Positives*) correspond to the number of instances correctly identified by the model as sharing a relation, TN (*True Negatives*) correspond to the total number of instances correctly identified as not sharing a relation, FP (*False Positives*) correspond to the total number of instances incorrectly identified as sharing a relation and FN (*False Negatives*) correspond to the total number of instances incorrectly identified as not sharing a relation.

Table 2.2 shows the contingency table for a binary relation classification task. When the true label is different from the prediction label, then FP and FN errors appear (for positive and negative relation types, respectively). On the contrary, in case of correct predictions, no errors appear and only TP and TN instances are counted. By making use of these statistics one can define evaluation metrics to estimate the performance of a relation extraction system.

		Ground Truth	
		Relation	No Relation
Prediction	Relation	**TP**	**FP**
	No Relation	**FN**	**TN**

Table 2.2: Contingency table of true positives (TP), false positives (FP), true negatives (TN) and false negative (FN) for binary relation classification.

Accuracy. Accuracy is used to estimate the number of systematic errors made by a

[c] https://civicdb.org/

model, i.e. a metric to estimate how *accurate* a model can be, in terms of how close the model predictions are to the truth. A combination of all the above statistics forms the accuracy equation,

$$\text{ACC} = \frac{TP+TN}{TP+FP+TN+FN} \tag{2.1}$$

However, accuracy fails to show the actual types of errors that the model is prone to make. It can also produce misleading high scores in cases of large imbalance between the target relation categories. In relation extraction datasets usually the number of negative relations are significantly more than the positive relations. As a result accuracy is not considered the best evaluation metric for such problems and two other metrics are used, namely Precision (P) and Recall (R).

Precision & Recall. Precision calculates the percentage of positively predicted (TP + FP) examples that are correctly predicted as positive (TP). On the other hand, *Recall* calculates the percentage of true positive examples (TP + FN) that are correctly predicted as positive (TP), hence measuring the sensitivity of the model.

$$\text{P} = \frac{TP}{TP+FP} \tag{2.2}$$

$$\text{R} = \frac{TP}{TP+FN} \tag{2.3}$$

Precision-Recall (PR) Curve. These two metrics can be plotted as a curve, describing the classification ability of a model based on various decision thresholds. The PR curve is created by typically plotting the Precision on the *y*-axis and the Recall on the *x*-axis. The Area Under the Curve (AUC) score which, in the case of a PR curve, is called *average precision* score, is used as an evaluation metric for relation extraction systems in cases the approaches do not explicitly plot the curve.

F-score. The metrics can also be combined to estimate another metric, named F_β-score, which is the harmonic mean of the two (Rijsbergen, 1979). In case of Precision and Recall being equally weighted ($\beta = 1$), the metric is called F_1-score.

$$\text{F}_\beta = (1+\beta^2) \cdot \frac{\text{P} \cdot \text{R}}{\beta^2 \cdot \text{P} + \text{R}} \tag{2.4}$$

False Positive/Negative Rate. These statistical metrics are used to estimate the discrimination capability of a system. The False Positive Rate (FPR) corresponds to the probability of a model to classify a randomly selected negative example as positive. On the contrary, the False Negative Rate (FNR) is the probability of a model to classify a randomly selected positive example as negative.

Micro- and Macro-averaged Metrics

The above described metrics are typically used in cases of *binary classification*, where the number of relation categories is equal to two. However, in the case of *multi-class classification*, one has to re-define both the statistics and the evaluation metrics. Firstly, the contingency matrix is modified with different errors for different relation categories. A toy multi-class classification example is described with three relation types (including the negative relation type) in Table 2.3. As observed, in the case of false predictions, the errors are summed for both the true relation and the predicted relation. For instance, if the prediction of a pair is B but the true label is A, then two types of errors are counted: a FP for category B due to misclassification and a FN for category A due to failure of prediction. In the case of prediction as *No Relation*, it is common to ignore the FP errors and only measure the FN errors for each missed relation category. Similarly, in case the true label is *No Relation*, only FP errors are counted for the each falsely predicted category.

		Ground Truth		
		Relation A	Relation B	No Relation
Prediction	Relation A	**TP** (A)	**FP** (A) & **FN** (B)	**FP** (A)
	Relation B	**FP** (B) & **FN** (A)	**TP** (B)	**FP** (B)
	No Relation	**FN** (A)	**FN** (B)	**TN**

Table 2.3: Contingency table of true positives (TP), false positives (FP), true negatives (TN) and false negative (FN) for a toy example of multi-class Relation Extraction with two relation categories (A and B).

In order to measure the performance of the model in a multi-class setting, the P, R and F1 metrics are transformed into micro- and macro- averages over the relation types. A *macro-averaged* metric computes the primary metric independently for each relation type and then estimates their average. On the contrary, a *micro-averaged* metric first aggregates the statistics for all relation types and then estimates the final score based on the aggregated statistics. The micro- and macro-averaged metrics for Precision and Recall are described in the following equations, where c indicates a relation category.

F-score is estimated in a similar manner by replacing P and R in Equation (2.4) with P_{micro}, R_{micro} or P_{macro}, R_{macro} respectively.

$$P_{micro} = \frac{\sum_c TP_c}{\sum_c TP_c + \sum_c FP_c}, \quad (2.5) \qquad R_{micro} = \frac{\sum_c TP_c}{\sum_c TP_c + \sum_c FN_c} \quad (2.6)$$

$$P_{macro} = \frac{1}{|c|}\sum_c P_c, \quad (2.7) \qquad R_{macro} = \frac{1}{|c|}\sum_c R_c \quad (2.8)$$

For the Average Precision score, as the area under the PR-curve, in multi-class classification problems, the PR-curve of each class can be aggregated into a single score using micro- or macro-averaging. The same stands for the FP and FN rates that can be calculated for all categories at the same time (micro-averaging) or for each class separately and then averaged (macro-averaging).

2.3 Taxonomy of Approaches

In the previous sections we categorised relation extraction based on a set of aspects that together can form different relation extraction tasks. Another division to follow is based on existing approaches developed for these tasks. In this dissertation, we attempt to classify relation extraction methods in four large categories. Our proposed categorisation is illustrated in Figure 2.5.

The first category, which is also the most common division followed by existing surveys (Bach and Badaskar, 2007; Zhang et al., 2017a), is based on the type of learning. Here, we define **learning** as the method used inside the model to exploit the provided textual data, either raw, human or automatically annotated. This category can be sub-divided into different learning approaches, including learning only from annotated data (supervised), learning by leveraging additional unlabelled data (semi-supervised), learning using distant signals from external resources (distantly-supervised), methods leveraging data from rich domains for prediction in lower resource domains (transfer learning), as well as methods that do no require any amount of labelled data (unsupervised). Learning from annotated data involves approaches that take advantage of the entire existing annotated set, as well as approaches that aim to learn using only a small portion and the annotated examples.

The second category is based on the **computational component** that each approach uses. Approaches can be categorised in statistical and neural. The former typically make use of explicit features to represent relation instances and feed them

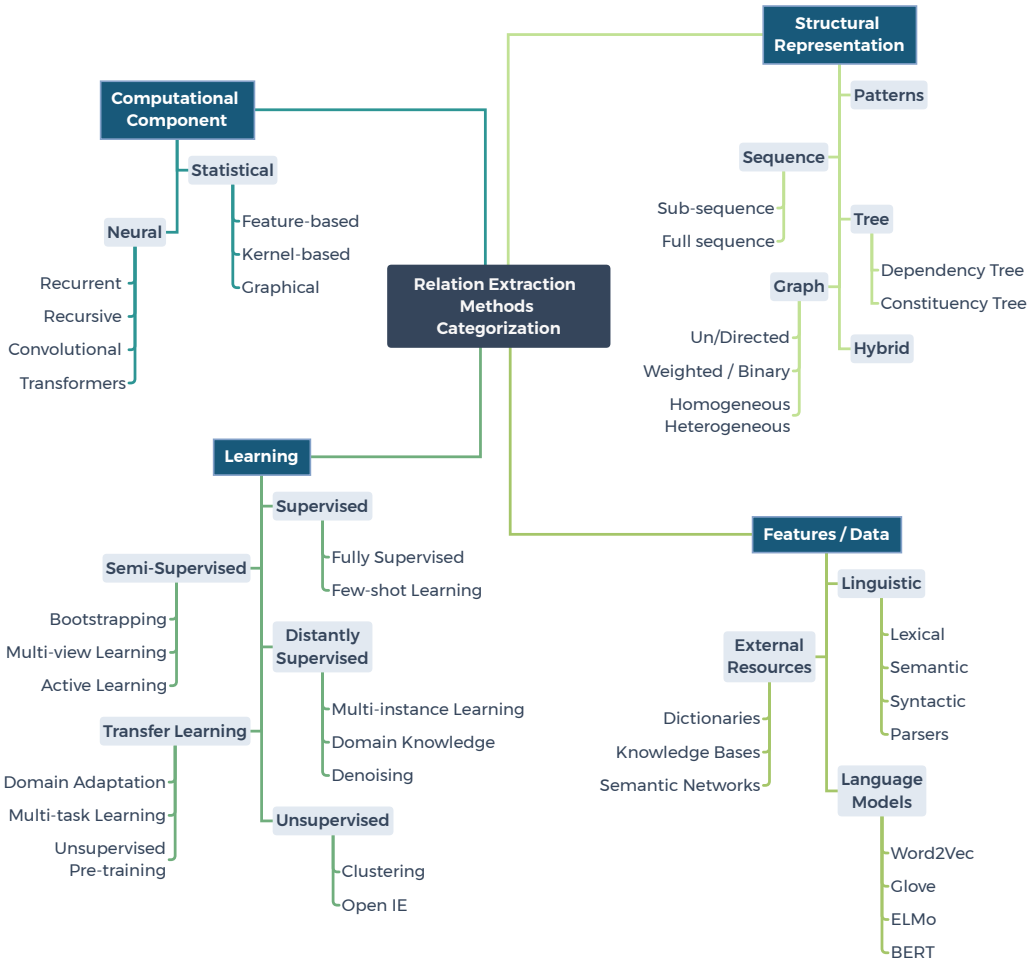


Figure 2.5: Taxonomy of Relation Extraction approaches.

into machine learning classifiers, use kernel-functions to compute similarities between different structures or probabilistic graphical models that treat the task as tracking probabilistic dependencies between random variables. The latter takes advantage of recent advances in neural networks, which represent instances with fixed-size, dense-vector representations, namely embeddings. Several architectures were proposed over the years (Recurrent, Recursive, Convolutional and Transformer networks), achieving state-of-the-art results.

Another division considers the type of the underlying **structural representation**. Existing approaches can be categorised in pattern-, sequence-, tree- or graph-oriented, though, as it will be revealed later on, most recent and best performing approaches actually employ a combination of different structures. Models have dealt with structure either implicitly (in the form of features) or explicitly (as transforming the input into

a certain structure), with kernel approaches introducing various structures for relation extraction, which were then adapted into neural approaches.

Finally, we can divide methods based on the type of **features** they incorporate, which is correlated with implicit structure representation. Typical features include linguistic features, features obtained from external resources such as Knowledge Bases or semantic inventories, as well as automatically constructed features obtained from language models, primarily used by neural architectures. Networks such as Word2Vec (Mikolov et al., 2013) and Glove (Pennington et al., 2014) automatically learn rich word representations from raw textual sources. The most recent advance in language representation learning constitutes the development of deep bi-directional neural-based models, named Transformers (Vaswani et al., 2017), which are able to produce higher quality dense word representations that capture a large range of linguistic phenomena (Clark et al., 2019).

Since the proposed methods of this dissertation follow the fully supervised setting, we will give more emphasis to approaches that use this type of learning. However, for purposes of completeness, we will briefly describe other learning settings including distantly-supervised, semi-supervised, transfer learning and unsupervised. We further split supervised relation extraction approaches based on the underlying textual structure they use, so that it is easier to later on categorise the approaches proposed in this thesis. Inside these categorisations, we describe the computational component used by each method, the target task and domain, as well as whether it relies on a particular set of features. We further point the reader to more detailed survey papers for more information regarding the corresponding methods.

2.3.1 Supervised Learning

Supervised learning methods treat the task of relation extraction as a classification problem, where the goal is to classify a set of arguments into a particular set of pre-defined relation categories. The timeline of supervised approaches is quite clear; feature-base methods were extensively used in the past, which were later replaced by kernel-based methods that were, in turn, overthrown by neural architectures.

Feature-based methods require that the input instances are transformed into feature vectors, with only the explicit information of these fixed-size vectors being used for classification, something that required the careful selection of features. Kernel-based methods (Vapnik, 1999; Cristianini et al., 2000) were employed to represent instances

in a latent space (typically of a higher dimensionality) in order to increase the expressiveness by investigating implicit feature spaces. A kernel function takes a pair of data as input and computes a score of similarity between them. The advantage of kernel-based approaches is that they can compute the dot product between two high dimensional vectors without explicitly mapping the vectors into the new space (Bishop, 2006). As a result, kernel-based methods followed a more structured approach by accepting structured input such as sequences, trees or graphs.

Since both feature-based and kernel-based methods in fact required a set of manually generated features as input, neural networks quickly replaced them. Their advantage compared to the former is, primarily, their property to contain richer information into a single vector representation of a fixed dimensionality. Neural models are the most recent advance in terms of computation, which came to replace kernel-based models by using rich representations, instead of hand-crafted features, to represent words. Many of the ideas obtained in the past, with feature- and kernel-based methods, were adopted by Neural Network architectures in an effort to represent text with sufficient structure and, at the same time, extract as much information as possible.

We attempt to classify both statistical and neural-based approaches into five categories that relate with the underlying text structure chosen by each approach. Although this classification breaks the strict timeline of developed approaches, we aim to show that older structures are consistently used with more sophisticated computational components as NNs. This eventually led to the usage of even more complex structures, such as graphs and hybrid approaches, that utilise a combination of multiple structures in order to represent relations.

2.3.1.1 Pattern-oriented Methods

Patterns were the initial form of structure used to extract relations from text. Approaches such as Agichtein and Gravano (2000) and Brin (1998) were using hand-generated patterns, either via soft or strict matching, to identify relations in text. A large resource of textual patterns for relations was also developed by Nakashole et al. (2012). Relations are organised in a taxonomy similarly to WordNet (Miller, 1995), harnessing information from existing Knowledge Bases. Such approaches, however, were quickly abandoned, particularly with the creation of human annotated datasets, since they were too restrictive in identifying multiple expressions that describe the same thing. Instead, methods moved on in representing text with other forms of structure such as sequences, trees or graphs, which latently use automatically generated

patterns.

2.3.1.2 Sequence-oriented Methods

The goal of sequence-based methods is to represent a set of arguments given their corresponding context as a sequence, considering word order. They take into account either the entire sequence or sequential parts of the input snippet, in order to identify the relation between a set of arguments.

Sub-sequence Modelling

Initial sequence-based approaches were based on kernels, with the method of Bunescu and Mooney (2006) proposing a sub-sequence kernel for extraction of interactions between Proteins from sentences. Their intuition lay in the observation that a relation between two entities is likely to exist in the sequence before, between or after the two entities in text. Their method showed promising results for extraction of protein-protein interactions; however, it failed to consider multiple interactions in the same sentence. Giuliano et al. (2006) adopted a similar approach for the same task by modelling the sub-sequences of the sentence similar to Bunescu and Mooney (2006), along with local context that is based on the target entities. The authors used only shallow linguistic features such as tokens, POS-tags, lemmas and orthographic features to represent entities, combining a local and a global context kernel linearly. They argued that the usage of other structured methods, such as trees, is restrictive since parsers were only developed for certain languages. An additional argument included the fact that trees ignored several information from the sentence. However, even stopwords can be useful for modelling the relation between a pair of entities, depending on the position they appear at. Results showcased that local context (information only about the entities) provided poor performance compared with global context, indicating that the entities alone are not enough to model their relation and richer information from the sentence context is required. A drawback of their approach was that they did not model the direction of the relation between two proteins, instead using the given direction from the data.

The rise of sequence-based methods was coincidental with the introduction of neural networks, especially networks that were developed to perform on sequences or sub-sequences. The first such models were Convolutional Neural Networks (CNNs), which emerged from their extensive usage on image processing. CNNs perform on short sub-sequences that are composed of a few continuous words in a textual snippet.

Initially, Collobert et al. (2011) proposed the usage of CNNs for a set of NLP tasks including POS-tagging, Named Entity Recognition, chunking and Semantic Role Labelling. The approach was adapted by Zeng et al. (2014), who were the first to propose CNNs for relation extraction. In their work, they underlined the extensive feature engineering required by previous approaches and proved that, by using CNNs, they could obtain significantly higher performance on relation classification without major pre-processing. One of their primary contributions was the introduction of relative distances from the target entities, which were embedding into dense vector representations. These representations were combined with word representations as input to the model, and proved to provide structural information regarding the position of the target entities in the sentence. Positional features provided a 10 percent boost in performance. Limitations of this approach include the fixation of positional embeddings during training to initially randomly assigned values, as well as the usage of a single window of size three, i.e. constructing only representations of tri-grams.

These limitations were later addressed by Nguyen and Grishman (2015), who proposed to use multiple windows in CNNs, consequently improving performance on relation classification. A further novelty of the approach was the introduction of pre-trained word embeddings with Word2Vec as the input to the model, in addition to enabling trainable positional embeddings. Moreover, they addressed the difficulty of training with imbalanced dataset, which is often the case in RE, showing that the number of negative instances greatly affects the performance of RE models. However, they do not model interactions between pairs in their approach and instead assume that only a single target pair resides in a sentence. dos Santos et al. (2015a) addressed the problem of the negative relation instances in relation extraction by forcing the model to focus more on the positive relation categories. They additionally showed that utilising pre-trained word embeddings obtained with Word2Vec, instead of distributional models, improves performance.

In the biomedical domain, CNN networks have been successfully applied on Drug-Drug Interactions (DDI) extraction (Liu et al., 2016) using multiple pre-trained word embeddings as input to the model, disease-treatment interactions (Sahu et al., 2016) and Protein-Protein Interactions (PPI) (Quan et al., 2016). However, in the work of Quan et al. (2016), the length of window sizes required in comparison to Nguyen and Grishman (2015), is much larger, potentially indicating that the associations between biomedical entities are further apart than those in the generic domain. As a result, the proposed approach failed to capture associations when the input sentence was very

long. Towards this direction, attention-based CNNs were developed in order to capture diverse contextual information with respect to the target pair, which may reside in different parts of the sentence. One such approach was proposed by Wang et al. (2016), where two attention mechanisms are incorporated to select relevant parts of the sentence for the target entity pair. The authors proposed an attention-based pooling function, instead of the max-pooling operation that was typically used in CNNs so far, achieving the state-of-the-art performance on the SemEval 2010 relation classification dataset. However, the authors point out that their approach fails to determine implicit relations, i.e. where there is not textual evidence of a relation, as well as pairs that are used in metaphorical contexts. Along the same vein, different CNNs were proposed for Drug-Drug interactions (Asada et al., 2017) and classification of treatment-disease relations in clinical records (Shen and Huang, 2016) by employing more sophisticated attention mechanisms.

Full Sequence Modelling

Despite the effectiveness that CNNs showed as sequential models for relation extraction, they had the disadvantage of not being able to model long-range dependencies well, as they focused on encoding local context using a short window. Although attention mechanisms helped in this regard, encoding the full context was impossible for these networks. This was something that could be solved by Recurrent Neural Networks (RNNs) (Elman, 1990) and their variants, as neural models developed particularly for sequences. The Long-Short Term Memory (LSTM) Network is a particular variant that showed to be effective in modelling long sequences. One of the first approaches to using Recurrent networks for relation extraction was that of Zhang and Wang (2015). The authors proposed a bidirectional Recurrent NN model that utilised only pre-trained word embeddings and position indicators instead of relative distances, as input to the model. The difference between the two is that position indicators are inserted as additional tokens in the sentence before and after each target entity. A max pooling operation accumulated the representation of the entire sequence into a single embedding. The proposed approach performed better than the CNN approaches with relative distance embeddings while evaluated on two generic domain datasets, and confirmed that Recurrent NNs are more effective in encoding longer sequences than CNNs. However, an existing limitation was that the method considered only one named entity pair per sentence. Zhang et al. (2015) proposed Bidirectional LSTM networks for relation extraction with relative distance embeddings, though they

incorporated additional features from parsers. They adapted the idea of Bunescu and Mooney (2006) into the sub-sequence kernel and split the input sentence into three spans; a max-pooling operation then generated a representation for every two consecutive spans. Dependency features boosted performance, though only the use of word embeddings as input feature yielded competitive results to CNN-based methods. The method was not directly comparable, however, to the simple recurrent variant, as the input word embeddings were different between the two methods. Kavuluru et al. (2017) augmented the input of an BiLSTM network with character-level information that was encoded with LSTMs, which were so far explored for other tasks such as POS-tagging (Santos and Zadrozny, 2014) and morphological language modelling (Kim et al., 2016). These features proved to be effective in the biomedical domain when extracting DDIs.

Attention-based approaches were proposed along with recurrent networks as well, such as the work of Zhou et al. (2016b). The authors used attention instead of max-pooling (as in Zhang and Wang (2015)) to produce the final sentence representation. The paper shows a detailed comparison among LSTM and RNN variants with different pre-trained word embeddings and position indicators. They showed that the difference between LSTM and RNN was not significant for the SemEval 2010 relation extraction dataset when using distributional embeddings. Attention mechanisms boosted performance by 2.5% and the usage of more informative word embeddings created by Glove or Word2Vec increased performance further, without incorporating other lexical features. This approach influenced subsequent works that proposed attention networks for detection of DDIs in the biomedical domain (Yi et al., 2017; Zheng et al., 2017; Sahu and Anand, 2018). A position-aware attention was proposed by Zhang et al. (2017c), along with a larger relation extraction dataset to satisfy the needs of deep neural models for large amounts of data during training (Adel et al., 2016). The authors underlined the problem of recurrent models in controlling the contribution of each word in the sequence explicitly, as well as the poor modelling of subject and object positions by prior work. Thus, they proposed to measure the importance of each word in the sentence via an attention mechanism that incorporates both the contextualised representation of the word, as well as the global position of the word in the sequence. In contrast with using BiLSTM networks, the authors used a single direction network and stacked two LSTM layers one on top of the other, which contributed to performance, but at the same time increased the computational cost. Similar to other approaches, however, the proposed model considered only a single pair per sentence, thus ignoring dependencies among

pairs in the same sentence. This was tackled in the approach of Sorokin and Gurevych (2017), who proposed an LSTM with attention to aggregate information from multiple pairs in the sentence when constructing the representation of a target named entity pair. The model considered the similarity of pairs in the same sentence rather than directly modelling their interactions.

2.3.1.3 Tree-oriented Methods

In contrast with sequence-oriented approaches, trees are more complex representations in terms of structure. However, they were one of the first representations used (after rules and patterns) in order to represent language and by consequence relations. This was mainly because syntactic parsing was an active research topic much prior to relation extraction; as such, there were already few syntactic and dependency parsers developed, that were able to provide tree-based structures for any textual snippet. Such approaches can be seen from two different perspectives. The first are approaches that incorporate dependency-based information derived from parsers as features into machine learning classifiers. The second are approaches which utilise a tree-oriented structure into the model directly. We attempt to describe both methods in this section. A strict separation of the former methods is not straightforward, since typically they constitute a combination of features. For this reason, we include herein approaches that are not restricted to tree-based features alone, but consider them in their representations. A drawback of such approaches, however, is their dependency on these external tools that severely restrict their domain portability, as well as propagate potential parsing errors inside the model.

Implicit Tree Structure

Implicitly tree structure was initially investigated by simply converting information from parsers into features. Kambhatla (2004) proposed a feature-based method that incorporated multiple features including syntactic, lexical and semantic ones. The additional features improved the expressiveness of the model compared only with using syntactic features. Later on, Zhou et al. (2005) demonstrated that base phrase chunking features can yield better performance compared with full parsing features, since shallow information contains the most informative parts in a sentence.

In the biomedical domain, Katrenko and Adriaans (2006) proposed a feature-based method with dependency tree features for the identification of PPIs and protein-gene interactions. Sætre et al. (2007) combined features from two different parsers into a

tree kernel. In their results, they proved that a combination of parsers yields better performance compared to using a single one. Trees were also used for extracting relations from Wikipedia Nguyen et al. (2007). Issues regarding domain portability of tree-oriented approaches were discussed in Miyao et al. (2008) for the task of detecting PPIs. In particular, the authors tested the output of various parsers by converting them into different tree- and graph-based representations. They showed that features from different parsers have different performance when re-training the parsers on data of a different domain, highlighting the need for domain adaptation when using such features, as well as portability issues of parsing tools. A similar concept was underlined by Liu et al. (2010). The authors pointed out that the words in a biomedical corpus are less discriminative for relation extraction compared to the words in newswire corpora, indicating the difficulty of RE in the biomedical domain. Furthermore, they proved that base phrase chunks are good features across domains and that dependency parsing features contribute more to the performance of biomedical corpora, where interactions are more complex.

Explicit Tree Structure

One of the first tree-based approaches utilising explicit tree structure was that of Zelenko et al. (2003), who used a tree-based kernel on constituency trees. They were able to take full advantage of the tree structure of a sentence using kernels, compared to methods that did not consider the structure of the text explicitly. Their proposed approach was, however, tested only on two relation categories while, at the same time, it was much slower compared to feature-based classifiers. Later on, Culotta and Sorensen (2004) proposed tree kernels on dependency trees instead of constituency ones. Their hypothesis was that similar relations will share a similar substructure in their dependency trees. For each entity pair they created an augmented tree, where each node was represented by a set of features. However, the model suffered from low recall and limited expressivity due to the requirement for matching nodes to be at the same tree depth.

The most well-known tree-based method was introduced by Bunescu and Mooney (2005), who proposed to use only the Shortest Dependency Path (SDP) in a dependency tree in order to represent a pair of entities. In fact, the authors constructed both a tree (using a Context Free Grammar parser) and a graph (using a Combinatory Categorial Grammar parser), but found that a kernel performing on the SDP of the sentence dependency tree yielded better results, due to the increased accuracy of the tree-based

parser. They proved that the SDP is enough compared to the common sub-tree used by Culotta and Sorensen (2004) and yielded significantly higher performance on the ACE 2003 dataset. Furthermore, since the representation was now a path instead of a tree, the kernel computation was much faster. Nevertheless, the method had much lower recall than precision since the compared paths required to have the same length, something that reduced the expressivity of the model. This finding was confirmed by Zhang et al. (2006a), who compared different parts of constituency trees for representing related pairs and concluded that, indeed, the shortest path in the tree connecting the two entities contains the most useful information and achieves the best performance with a convolution tree kernel. They further showed that semantic entity type features can contribute significantly to the detection of relations, something also confirmed in Zhang et al. (2006b).

The first approach that utilised neural networks in relation extraction was proposed by Socher et al. (2012) on the SemEval-2010 dataset. In particular, a Matrix-Vector Recursive Neural Network was introduced, that performed on the path of a constituency tree that contained the two target entities. The network formed representations for constituents in a bottom-up manner by using a compositionality function. The method influenced several subsequent methods to apply Recursive networks instead of kernels. Hashimoto et al. (2013) followed the same approach but used a different compositionality function, where phrases of importance were explicitly weighted more than non-informative ones. Later on, Ebrahimi and Dou (2015) proposed the application of a Recursive NN on the shortest path between two entities in a dependency tree compared with the previous models that performed on constituency trees. The advantage of the approach lay in the faster computation of the proposed architecture. In a similar vein, Xu et al. (2015c) also utilised the shortest dependency path between two entities on a dependency tree, but instead utilised a sequential LSTM network to encode it. They used two LSTMs along the left and right sub-paths of the SDP to encode information for the root node. A max pooling operation over the multiple representations for words, POS-tags, grammatical and WordNet features, resulted into a final representation for the root, which was then classified into a relation category. Liu et al. (2015) observed that methods that utilise SDPs between two relation instances can have a similar SDP but belong to different relation categories. As a result, methods employing only SDPs can fail to classify them correctly. They thus proposed to augment the SDP with the sub-trees attached to each node in the path, in order to form a more informative structure. A combination of recurrent and convolutional networks helped

construct the representation of a pair. They concluded that the two structures (SDP and sub-trees) play different roles in RE and appropriate architectures are required to encode this information. Since sequential and recursive models have been used, Li et al. (2015) analysed the effectiveness of Recursive versus recurrent networks and concluded that recursive architectures are more effective when dealing with arguments far apart in the sentence. However, the use of bi-directionally in recurrent networks reduces the performance difference between the two. A sequential model generalised to operate on tree-structured topologies was proposed by Tai et al. (2015) and evaluated on semantic sentence relatedness and sentimental classification. The method was adapted by Miwa and Bansal (2016), who combined sequence and tree representation in a hybrid model for simultaneous extraction of entities and relations. In their analysis, they concluded that the input structure (i.e. sequence or tree) is more important than the actual model used to encode it (tree LSTM or sequence LSTM), demonstrating the importance of structural information in relation extraction. Cai et al. (2016) introduced LSTMs to encode the SDP between a target pair and CNNs on top of them to extract local features from the dependency units. They applied this process to both directions and concatenated the two outputs.

In the biomedical domain, Wang et al. (2017) utilised LSTMs on tree-structures that were converted into sequences using Depth First Search (DFS) and Breadth First Search (BFS). Lim et al. (2018) utilised a novel position encoding scheme to better measure the relative distance between words in a sentence. They fed position and word embeddings into a tree-LSTM network and achieved the state-of-the-art performance in DDI identification. Drawbacks include failure to detect relations in complex sentences or when drugs are far apart from each other. Moreover, factuality or speculation are not handled properly, thus classifying pairs as false positives.

More recently, the rise of Graph Convolutional Neural networks (GCN) (Kipf and Welling, 2017) opened new ways to support the encoding of tree-structures for relation extraction. An initial approach was introduced in Zhang et al. (2018b), who used the SDP between two entities of the dependency tree of a sentence as input to a GCN network. A pruning strategy was incorporated in order to remove uninformative edges from the tree.

2.3.1.4 Graph-oriented Methods

In comparison with trees, graphs are more flexible structures that enable multiple types of interactions. A major difference between trees and graphs is that the former are restricted into having only a single parent node, whereas the latter enable arbitrary connections between nodes. Graph-oriented approaches use a graph structure and attempt to encode it via a machine learning algorithm. Existing approaches can be broadly divided into two categories: methods that simply perform on existing graph structures, i.e. the input graph is given to the model, and methods that first construct the graph and then use a particular model to encode the interactions in the structure.

Link Prediction

With regard to the first category, pre-defined graphs can roughly be considered to belong into two families: Knowledge Graphs and Networks. The former corresponds to the graphical representation of a Knowledge Base and has a strictly defined schema in the form of triples consisting of two arguments and a relation between them. Relations come from an existing set, while arguments belong to particular semantic categories (typically fine-grained), such as people, lawyers, locations, and so on. On the contrary, Networks refer to structures that model relationships between much broader categories; users in social networks, for instance. In addition, Knowledge Networks combine diverse information from multiple KBs, where the association between nodes in this case, is more loosely defined.

Link prediction in Knowledge Graphs is a task relevant to relation extraction. Contrary to typical RE, it deals with multi-relational data where the potential relation categories can be thousands. Furthermore, relations are always extracted between named entity concepts rather than mentions. This means that KGs contain factual relations, i.e. relations that are not dependent on a specific context (e.g. a sentence), but instead generally hold as ground truth. Since all KGs are incomplete, the goal of KG link prediction is to find potential missing associations (edges/links) between concepts (nodes).

Compositional approaches are vector space models that aim to construct pair representations of particular relations using compositionality functions. Socher et al. (2013) proposed the neural tensor network with an augmented bilinear transformation. The model offered high expressiveness but required many parameters, something that was later improved by Nickel et al. (2016), who proposed holographic embeddings using

circular correlation between the representations of two concepts. Translational approaches aim to predict the existence of a relation based on the similarity of entity and relation embeddings in the vector space. The most well-known approach is that of Bordes et al. (2013), named TransE. The authors represent a relation as a translation operation between two concept vectors, inspired by the properties of Word2Vec (Mikolov et al., 2013) vectors. Several subsequent models attempted to improve expressiveness while preserving computational efficiency (Wang et al., 2014; Trouillon et al., 2016). Recent state-of-the-art approaches for link prediction utilise tensor factorisation (Balazevic et al.) or Graph Neural Networks (Schlichtkrull et al., 2018; Dettmers et al., 2018).

Link prediction has also been targeted for networks, using graph-based algorithms that encode the nodes into low dimensional representations taking advantage of the structure of the network. One of the most used algorithms is DeepWalk (Perozzi et al., 2014), which was motivated by the SkipGram architecture of Word2Vec (Mikolov et al., 2013) and adapted the idea into modelling a stream of short random walks on a graph as a sequence. LINE (Tang et al., 2015) investigates both first and second-order proximity neighbourhoods. The authors proposed an edge-sampling strategy which scales linearly to the number of edges and is independent of the number of nodes. Finally, Node2Vec (Grover and Leskovec, 2016) is an improved extension of DeepWalk that constructs node representations based on flexible neighbourhoods in the network, by using a combination of both Bread-First-Search (BFS) and Depth-First-Search (DFS) sampling strategies in graphs.

In this thesis, we borrow a few ideas from compositional approaches that have been used in KG link prediction, as will be discussed in Chapter 4. Since, however, link prediction is not our target task, we refer readers to more detailed surveys for models developed for link prediction in KGs (Nguyen, 2017) and networks (Martínez et al., 2017; Cui et al., 2018b).

Graph Construction and Encoding

As seen in the Section 2.3.1.3, there are models that apply graph-encoding mechanisms on tree-based structures obtained from dependency parsers (Zhang et al., 2018b). In general, approaches that construct graph structures and then encode them, typically rely on trees obtained from a parser, and utilise heuristics in order to transform them into graphs. As we will see, the majority of such approaches have been proposed for cross-sentence relation extraction.

A first approach was proposed in McDonald et al. (2005) where *n*-ary relations were decomposed into a set of binary associations represented as a non-directed graph, with entities as nodes and associations between them as edges. A classifier is trained on the binary associations and then maximal cliques are selected from the generated graph to form *n*-ary associations. In particular Bordes et al. (2013) proposed to model shortest paths on graph constructed via dependency for PPIs. Later approaches on graphs were kernel-based, with Airola et al. (2008) proposing a graph kernel for PPIs. They proposed a graph representation for each candidate relation instance, taking into account the shortest dependency hypothesis of Bunescu and Mooney (2005), the linear order of the sentence and all the paths between two target entity nodes in the graph. They showed that other kernel methods lack the expressive power to consider complex representations that create cycles, such as graphs. Most recently, Panyam et al. (2018) proposed two graph kernels that perform on enhanced dependency graphs with edge weights. This enables the model to distinguish between the SDP in the graph and other, likely informative, paths. However, the method failed on sentences that contained multiple entities.

A first approach for cross-sentence relation extraction in the biomedical domain was proposed by Quirk and Poon (2017). The authors underline that the limited amount of work in cross-sentence RE is attributed to the domain of focus. As they explain, sentences containing relations of interest are more likely to exist in the newswire domain; a rarer occurrence in the biomedical domain, where inter-sentence interactions can have significant impact for interaction discovery (Banko et al., 2007). The method proposed a document-level graph, with words as nodes and edges constructed from dependency parsing, co-reference resolution and discourse relations. Additional heuristic adjacency edges between adjacent words were used in order to tackle potential errors from the parser. They incorporated a binary logistic regression classifier and considered K-shortest dependency paths between the two entities as input to the model. Evaluation showed that coreference edges were of poor quality for the biomedical domain and thus reduced performance. A drawback of the approach was that it only considered associations that span up to three sentences.

A subsequent approach by Peng et al. (2017) extended the document-level graph for detection of *n*-ary relations in the biomedical domain. The authors followed the same setting and data proposed by Quirk and Poon (2017), but instead used a graph-LSTM network on two Directed Acyclic Graphs (DAG) for forward and backward dependencies, respectively, which was an extended version of the tree-LSTM proposed

in Miwa and Bansal (2016). They showcased that LSTM networks performed better compared to CNNs for binary, cross-sentence RE, and graphs performed better than tree-based approaches on the SDP. They additionally proved that multi-task learning between ternary (three arguments) and binary relation extraction improved the performance of both settings regardless of the usage of sequential or graph LSTMs. Similar to (Quirk and Poon, 2017), co-reference and discourse edges obtained from parser offered no significant gains, but when using gold dependencies in the document graph results were improved. Later Song et al. (2018) improved the idea by modelling the entire document graph as a whole using a recurrent graph network. The authors underline the lower performance on intra-sentence RE is due to potentially less training examples, as well as the fact that single sentences do not provide enough context for cross-sentence RE, thus information from other sentences is needed to enhance their representations.

Recently Zhu et al. (2019) introduced another GCN network for relation extraction by incorporating edge information into the node representations in order to improve detection of multi-hop relations. In their approach, entities are considered to be nodes. The connection between each pair is encoded separately by using an LSTM network into an edge weight. Guo et al. (2019) extended the idea of Zhang et al. (2018b) and incorporated an attention mechanism into GCN in order to prune the trees, instead of using rules. The model was applied on full dependency trees which were transformed into fully connected graph structures. The method is a soft edge pruning approach and was able to achieve state-of-the-art performance in binary and n-ary RE inside and across sentences. Fu et al. (2019) introduced an end-to-end model for extraction of relations based on a bidirectional GCN network on directed graphs. The authors proposed a two-step procedure in order to further take advantage of interactions between named entities and relations and were able to successfully model overlapping relations, i.e. relations that share at least one entity. They showcased that modelling interactions between relations came with significant performance improvements. The model was applied on silver corpora, which are not human annotated. Sahu et al. (2019) proposed a GCN encoder for inter-sentence relation extraction over syntactic dependency graphs. The difference in the approach in comparison with other methods that used trees to create graphs, was that they incorporated the Enju dependency parser who extracted predicate-argument structures, thus directly resulting into a graph. Co-reference and adjacency edges were added following previous work, though coreference did not offer improvements. As one of the latest advances, Zhao et al. (2019)

combined GCNs with BERT (Devlin et al., 2019) to model associations between pairs of entities, using shared entities across sentences, thus encoding the topology of the graph. Contrary tp approaches where the nodes of the graph corresponded to words, the nodes in this case corresponded to entities. This was something that was also adapted for graph construction from other approaches as well, such as De Cao et al. (2019) for cross-document question answering. Constructed entity-word graphs were also explored by Tagawa et al. (2019) for knowledge graph completion.

2.3.1.5 Structural Hybridity

In reality, the majority of the aforementioned structure-based methods are hybrid. Hybrid approaches can include a mixture of different models that operate on the same structure; for instance, using different syntactic kernels (Zhao and Grishman, 2005) or multiple sequential models (Vu et al., 2016; Zhou et al., 2017; Raj et al., 2017). Secondly, they can be a combination of algorithms that perform on different structures. For example, the work of Miwa and Bansal (2016) clearly belongs to this category, as it explicitly encodes and combines sequences with trees. Moreover, Zhang et al. (2018a) divided the SDP between entities into a dependency word sequence and a relation sequence. They used different networks to encode different structures. Early analysis on combining structures can be found in Jiang and Zhai (2007), who investigated the difference in performance when using sequence, syntactic parse tree or dependency parse tree features. They found that each subspace has adequate information for representation of relations. When combining features from different levels of complexity with additional feature pruning the best performance could be achieved. This was probably due to the combination of diverse information from the different structures.

Furthermore, latest work on graph structures is mostly hybrid. For instance, the model of Zhang et al. (2018b) may explicitly use tree structures, though their best model uses contextualised representations of nodes, which are constructed using a BiLSTM network on the input sentence. In essence, sequential information is used implicitly in the tree structure. Similarly, the extension work of Guo et al. (2019) performs best with contextualised node representations. In their work it is stated that the combination of densely connected (graph) structures and attention on tree structures is able to produce better representations for downstream tasks.

In Table 2.4, we provide a summary of the general benefits and drawbacks of existing structure-based approaches for relation extraction. As observed, graphs combine

	Benefits	Drawbacks
Pattern	• High accuracy	• Restrictive expressiveness • Low recall
Sequence	• Consider discourse order • Capture local/global discourse information	• Ignore complicated associations • Consider uninformative words in the sequence
Tree	• Ignore non-important words • Perform well for long sequences	• Dependent on domain-specific tools • Structure does not imply linear ordering
Graph	• Large expressiveness • Can be constructed without domain-specific tools • Benefits of trees and sequences	• Search space can be too large

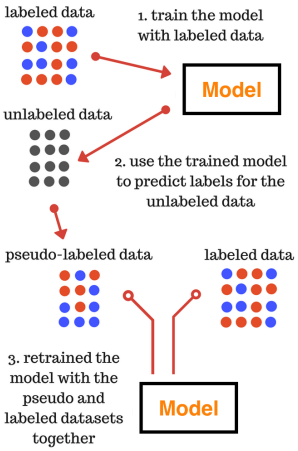
Table 2.4: Table summarising benefits and drawbacks of methods using different structures.

advantages from both sequences and trees, offering larger flexibility and expressiveness. Recent advances in relation extraction using contextualised graph-based structures, further motivate our ideas for graph-based techniques for detection of not only intra- but also inter-sentence associations.

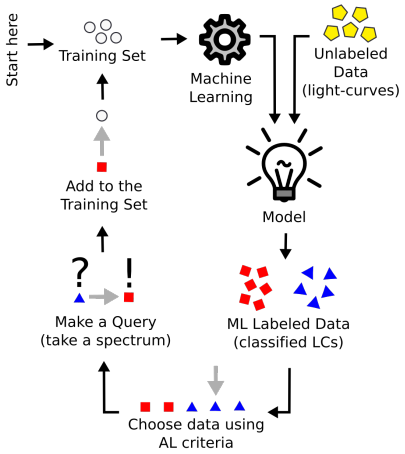
2.3.2 Semi-supervised Learning

The generation of gold annotated data is an expensive and time-consuming process for humans. Semi-supervised learning methods aim to take advantage of the large amount of existing unlabelled data in the web, and use them in conjunction with existing, but small, human-annotated data. One way to perform semi-supervised learning is to rely on trained algorithms that enable the automatic labelling of unlabelled data.

Bootstrapping is one of earliest and most common semi-supervised techniques. An abstract framework of is depicted in Figure 2.6a. Initially, a set of seed examples (existing annotated instances) are utilised to assist in the annotation of additional data. Examples annotated with high confidence are integrated into a larger set of labelled data used to re-train the model. One of the first bootstrapping systems for relation extraction was DIPRE by Brin (1998) for extraction of *author-book* relations from the Web. The method was pattern-based, thus suffered from limited expressiveness



(a) General intuition about semi-supervised learning. Source: www.analyticsvidhya.com



(b) General procedure of active learning. Source: http://inspirehep.net

and low recall. A similar approach is that of Agichtein and Gravano (2000), where they identified *organisation-location* pairs with their system, *Snowball*. They firstly represented each pair as a pattern-vector and then grouped vectors into classes using semantic similarity, resulting in better recall. Later, Xu et al. (2007) studied n-ary relation extraction with bootstrapping in their system *DARE*, again using rules. Other bootstrapping approaches include relations from the web (Pantel and Pennacchiotti, 2006), utilisation of co-reference (Gabbard et al., 2011) and and usage of probabilistic models (Pawar et al., 2014). Studies such as Vyas et al. (2009); Kozareva and Hovy (2010) provide an extensive investigation on different criteria for selecting seed sets for bootstrapping.

Although bootstrapping is an efficient method, it typically suffers from semantic drift. Another set of approaches to tackle this use multi-view learning, which essentially trains two different classifiers on different views of the data. Training enforces agreement between the two classifiers before selecting data to augment the training set. One such algorithm is co-training (Blum and Mitchell, 1998). Zhang (2004) represented a relation instance using lexical and semantic features. In their co-training algorithm, they randomly project the feature space into different views with random feature projection. Multiple classifiers trained on the feature projections produced votes for the inclusion/exclusion of unlabelled examples into the training set. Similarly, Li et al. (2016b) trained a feature and a graph-based kernel on two different views of the data for intra-sentence biomedical relation extraction.

A different set of methods utilise active learning as a way to label unlabelled data.

Active learning is a kind of learning algorithm where a machine learning system iteratively learns and selects a small, but informative amount of unlabelled data, gives them to human annotators and then adds them to the training data (Figure 2.6b). Following this method, Mohamed et al. (2010) studied different ways to select informative data for labelling by developing four different strategies. They applied their method on PPI detection. Zhang et al. (2012) proposed an active learning technique for biomedical relation extraction with several stages. They studied how to select informative data for labelling, how to remove duplicate examples between annotators, effective feature generation for each example and, finally, efficient feature selection for the learner. Sun and Grishman (2012) used active learning with multi-view learning using local and global views of a relation instance. Later on, Fu and Grishman (2013) improved the model in terms of efficiency.

2.3.3 Transfer Learning

In comparison with other learning schemes, Transfer Learning (TL) is used when the labelled data in a certain domain are relatively few or non-existent. Thus, the goal is to leverage information from auxiliary domains (domain adaptation) or tasks (multi-task learning) in order to make predictions for the target domain. A recent trend, however, is unsupervised pre-training of language models that produce high quality word embeddings, which can be incorporated directly into existing neural architectures.

Initial approaches used rules (Feiyu Xu and Felger, 2008), while others used kernels combined with lexical information generalised by clustering or similarity (Plank and Moschitti, 2013). The latter method showed that the usage of Brown clusters (Brown et al., 1992), combined with tree-based kernels, offered significant improvement when no available labelled data exist for the target domain. At the same time they confirmed previous findings that gold semantic entity type information into the model, offer significant improvements. Word embeddings and clustering features were also investigated by Nguyen and Grishman (2014), using a hand-crafted feature model under the same setting. On the contrary, Nguyen et al. (2014) follow a different setting, where a few labelled data exist for the target domain. A subsequent approach showed that combining word embeddings into tree-kernels yielded the best results (Nguyen et al., 2015), while later Fu et al. (2017) showed that incorporation of neural networks with adversarial training further improved performance without hand-crafted features. A similar intuition was proposed by (Rios et al., 2018); although the method is promising, it suffers from the class imbalance problem during training.

Another set of approaches for TL include multi-task learning. The work of Jiang (2009) relied on the intuition that different relations can have common structures and suggested a combination of entity type constraints and feature generality via human knowledge. Thus, multi-task learning between different relation categories could help in transferring knowledge, using a weight vector to share features among different domains. The latest approach of Sanh et al. (2019) uses a neural model on several semantic tasks, including NER, co-reference resolution and relation extraction. Their intuition is that some tasks are low-level and relatively easy to solve, while others are high-level, meaning that they require deeper processing. Their hierarchical multi-task learning approach achieves state-of-the-art performance on all tasks.

The latest trend in TL approaches is unsupervised pre-training, by training language models (Devlin et al., 2019; Peters et al., 2018) in massive raw corpora and then fine-tuning them on other domains that have limited labelled data. The difference with early work on language modelling is that these models can capture more complex linguistic phenomena (Clark et al., 2019). Improvements using these approaches have been shown for both the generic (Baldini Soares et al., 2019) and the biomedical domain (Peng et al., 2019).

2.3.4 Distant Learning

Distant Supervision (DS) for relation extraction emerged as a different learning setting that does not require any human annotated data, as is the case for supervised and semi-supervised learning. In contrast, labelled data are obtained automatically using distant signals, typically from existing Knowledge Bases that contain a large amount of relational facts. Although this relaxes the need for human annotation efforts, the procedure that must be followed is not that simple. In particular, one must first perform named entity recognition on the raw text and then apply entity linking (Hachey et al., 2013) between the textually identified named entities the ones existing in the KB. It is then possible to align the raw text with the KB information and automatically annotate large amounts of unlabelled data. The initial scheme of the distant annotation process is shown in Figure 2.7. If relations exist between named entities in the KB, then sentences containing these entities can potentially be labelled with the KB relations.

Distant Supervision has been broadly used during the last decade for automatic annotation of relation extraction datasets. First Mintz et al. (2009) proposed a simple DS setting, based on the following assumption:

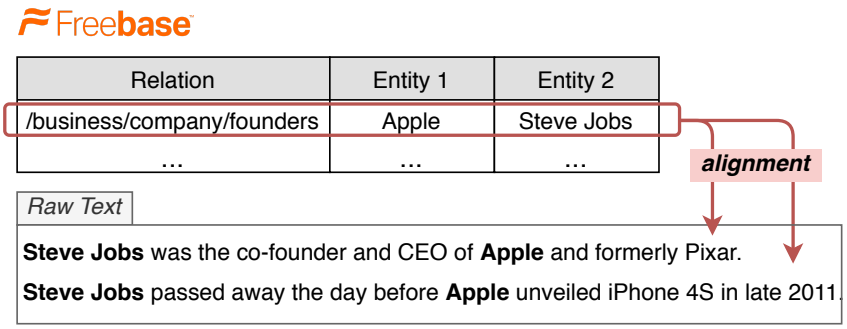


Figure 2.7: Abstract schema of Distant Supervision (DS) annotation procedure. Image adapted from Zeng et al. (2015).

"If two entities participate in a relation (in the KB), any sentence that contains those two entities might express that relation.".

In essence, if two entities co-occur in a sentence and are also annotated as related in the KB, they are assigned a positive relation label from the KB. If two entities co-occur but are not annotated as related in the KB, they are then given the "no relation" label. The authors utilised Freebase to annotate Wikipedia articles in order to collect a large set of distantly annotated sentences.

However, this automatic annotation process can introduce a lot of noise (false positives) by assigning relations to pairs that are not actually related given a particular context. To deal with the noisy corpora that this setting produced, Riedel et al. (2010) introduced Multi-Instance Learning (MIL). As the authors mention in their paper, "When the knowledge base is external, entities may just appear in the same sentence because they are related to the same topic, not necessarily because the sentence is expressing their relations in our training knowledge base". The MIL setting relies on the *expressed-at-least-once* assumption as follows:

"If two entities participate in a relation, at least one sentence that mentions these two entities might express that relation."

In this setting, the input is considered to be a bag-of-sentences, i.e. all sentences associated with a certain pair. The argument types are concepts, rather than mentions, and classification is performed for each concept pair (fact), considering each pair occurrence in a sentence as another mention of the pair. Each model is then responsible for selecting informative instances from the bag in order to identify whether a pair shares a certain relation or not. In the approach of Riedel et al. (2010), the authors aligned Freebase with the New York Times (NYT) corpus in order to achieve complete

separation between the raw text and the KB, since Wikipedia shares many data with Freebase. They developed a graphical model to distinguish whether two entities are related in a certain sentence.

However, even in this setting, each pair of interest is assigned a single relation label. Hoffmann et al. (2011) argued that this scenario is too restrictive and, in reality, a target pair in a KB can share multiple relation types, i.e. multi-label classification. They proposed a novel graphical model that allows both multiple instances of a target pair and overlapping relations. In a similar manner, Surdeanu et al. (2012) formally addressed multi-instance and multi-label RE on distantly supervised corpora using graphical models. However, since KBs are incomplete, i.e. do not contain all possible facts of the world, false negatives are also an issue, as pointed by Min et al. (2013). As a result, most methods report the performance in terms of Precision-Recall curves on held-out data, or report the precision on the top K most confident pairs against human annotations.

Later on, several approaches were developed to deal with distantly supervised corpora, using multi-instance learning following the *expressed-at-least-once* assumption, and particularly investigated how to select informative instances from the bag. Zeng et al. (2015) proposed to use a Piecewise CNN (PCNN) without relying on automatic NLP tools for text processing and feature engineering (e.g. parsers). However, in their method, they only selected one sentence from the bag, leading to large information loss. Attempts to solve this limitation were proposed by several methods, including cross-sentence max pooling over the sentences of the bag (Jiang et al., 2016), CNNs with attention over the bag instead of max pooling (Lin et al., 2016), ranking losses (Ye et al., 2017), hierachical attention to select only a few sentences (Zhou et al., 2018) and structured attention of words and sentences (Du et al., 2018).

Another set of approaches, targeted to incorporate additional information from the text or the KB (domain knowledge). Zeng et al. (2017b) incorporated indirect information from other sentences in the dataset that contain at least one of the target entities of a pair. Differently, Ji et al. (2017) used a PCNN with attention over sentences by additionally incorporating entity descriptors to directly add KB knowledge information into the entity representations. Vashishth et al. (2018) proposed to include relation aliases and semantic entity types obtained from the KB, further enhancing distantly supervised RE. In a similar vein, She et al. (2018) utilised entity descriptors from Wikipedia for English and Chinese relation extraction. There is also a very recent amount of work that utilises information from the KB in the form of embeddings (Wang et al., 2018;

Xu and Barbosa, 2019; Zhang et al., 2019a; Trisedya et al., 2019) which is very much aligned with link prediction and KB enrichment.

A different group of methods focuses more on the noisy nature of distantly created data and proposes measures to effectively reduce it. These methods can also be seen as attempting to reduce the false negative instances during training. Early methods include the work of Takamatsu et al. (2012), who introduced a generative model to predict wrong relational patterns and removed instances based on them. Xu et al. (2013) proposed a passage retrieval model based on the assumption that pairs found in many relevant sentences to a particular relation type, are likely to express the relation. Zeng et al. (2017a) developed a cost-sensitive ranking loss to deal with class imbalance, while other approaches performed filtering using heuristics (Intxaurrondo et al., 2013), cluster-based sampling (Sterckx et al., 2014) and reinforcement learning (Feng et al., 2018). Following a different approach Ren et al. (2017) proposed a model that deals with the noisy nature of the data with the help of semantic similarity between the representations of entities and relations. Luo et al. (2017) tried to directly model the noise distribution in DS data and use it in a transition matrix to produce estimations concerning the noise in the predicted labels. Similarly, Qin et al. (2018) introduced a generator-discriminator model for noise reduction in DS corpora. DS has been effectively introduced for annotation of documents or paragraphs (including intra- and inter-sentence relations) for the generic (Web) (Augenstein et al., 2014) and the biomedical domain (Quirk and Poon, 2017; Verga et al., 2018). For a more detailed review of DS methods we refer readers to the extended survey of Smirnova and Cudré-Mauroux (2019).

2.3.5 Unsupervised Approaches

In contrast with all previous approaches, unsupervised methods do not require any labelled data from any domain (source or target) or human curated Knowledge Bases. These methods generally fall into two categories: Clustering methods to induce relation types and Open Information Extraction (OIE) from text. The difference of classic RE with unsupervised clustering methods is that, in the latter, relation labels are essentially clusters, thus enabling the discovery of new relation categories. On the other hand, Open Information Extraction finds possible triples in the form of *subject*, *predicate*, *object* that express a relation from text, as shown in Figure 2.8.

Most existing clustering approaches rely on measuring semantic similarity or co-occurrence. Hasegawa et al. (2004) first proposed clustering for unsupervised RE

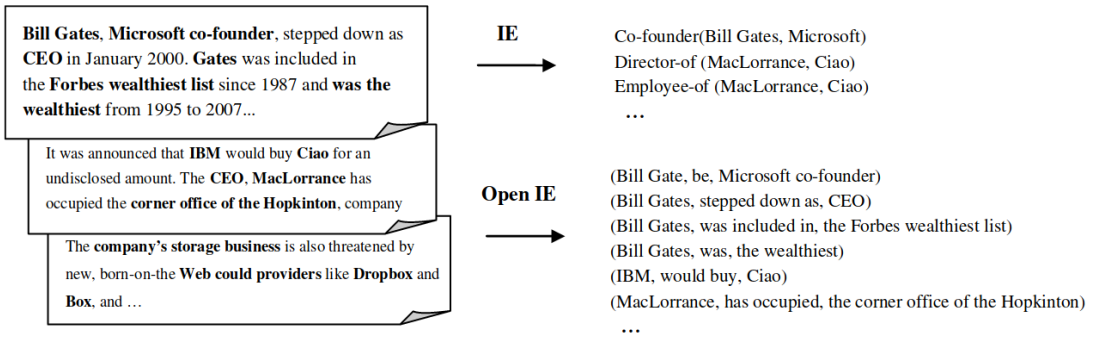


Figure 2.8: Relation examples extracted from typical information extraction systems and Open Information Extraction systems.

gathering co-occurring named entities in sentences along with their context words (in-between the two named entities). Chen et al. (2005) improved this approach by using feature selection criteria during clustering. Additionally, they automatically estimate the number of clusters and utilise discriminate category matching for better identification of relation labels from clusters. Yan et al. (2009) proposed relation extraction between Wikipedia concepts, where pairs were clustered together based on the similarity of their contexts. Lin and Pantel (2001) relied on the distributional hypothesis (Harris, 1954) that words that co-occur together tend to be similar. In this work, the authors conclude that "paths in dependency trees have similar meanings if they tend to connect similar sets of words". They generated inference rules and discovered potential relations from unstructured text. Later on, Yao et al. (2011) experimented with generative probabilistic models on raw corpora. They used a variation of the LDA algorithm (Blei et al., 2003) for topic modelling, where the relation categories were viewed as latent topics. In quite a similar manner, Lopez de Lacalle and Lapata (2013) utilise a topic model to extract an arbitrary number of relation categories from tuples representing syntactic relations between entities. More recent approaches utilise discrete-state variational autoencoders (Marcheggiani and Titov, 2016).

Open information extraction was firstly introduced in Banko et al. (2007) and their system TextRunner for OIE from the Web. Wu and Weld (2010) improved TextRunner's precision and recall by utilising Wikipedia infoboxes to generate better training data. Another well known OIE system is that of Fader et al. (2011), namely ReVerb. The authors focused on extracting identifiers for binary relations expressed by verbs in English and proposed syntactic and lexical constraints over them. Then, Etzioni et al. (2011) improved ReVerb's argument detection module by incorporating an additional

argument learner. Mausam et al. (2012) proposed OLLIE, an improved version of ReVerb that tackles two of its principal shortcomings. Firstly, it explores more relational phrases than verbs (such as adjectives and nouns) and, secondly, acknowledges context information into the relation extraction procedure. Gamallo et al. (2012) utilised a dependency parser to obtain rich information for arguments and relations in text. After dependency parsing, verb clauses are identified along with their participants. A set of rules was employed to extract relation triples.

Recently, Graphene (Cetto et al., 2018) was developed for sentences that have a complex linguistic structure. The model builds a two-layer hierarchical representation in the form of facts and their accompanying contexts. It additionally extracts rhetorical relations through which facts maintain their semantic relations. Niklaus et al. (2018) provide a detailed survey of several existing OIE systems. A first step towards neural OIE was taken by Cui et al. (2018a), where an LSTM encoder-decoder network was trained on the extracted relations of an existing OIE system. Their method showed promising results compared to state-of-the-art models for the same task. Recent methods for OIE try to take advantage of Question-Answering datasets and transform them into training data. In more detail, Stanovsky et al. (2018) treat unsupervised RE as a semantic role labelling problem using transformed QA datasets. They showed the benefits of coupling QA and OIE data to train better relation extractors. Following the same trend, Bhutani et al. (2019) suggested a neural model that combined vector representations of both questions and answers to extract knowledge facts. The system also utilised relevant information from multiple sentences, outperforming other models on predicting relation tuples from QA pairs.

2.4 Conclusions, Limitations and Challenges

In this chapter, we presented a broad overview of relation extraction. We firstly categorised RE into several associated tasks that are formed based on the kind of information one aims to extract. Investigation however, through related work, revealed that the majority of approaches have mainly targeted sentence-level relation extraction, while document-level relation extraction and, by consequence, relations of higher order (n arity) remain an understudied field, for both the generic and the biomedical domain. This further motivates our investigation for document-level RE, as will be discussed in Chapter 6.

We extensively discussed prior work in the field, where we divided existing methods into four large categories, the subcategories of which can fully describe an existing method for RE tasks. In particular, the type of learning was considered our first division into supervised, semi-supervised, distantly supervised, transfer learning and unsupervised techniques. We focused mainly on fully supervised techniques, since this is the main learning setting of our methods. Other learning techniques were also mentioned for the purposes of completeness, as well as to illustrate that significant efforts have taken place in order for models to generalise well without the dependence on learning from large amounts of human-annotated data.

Our second and more fine-grained division relied on the structural representation of each method. We partitioned these approaches into pattern-, sequence-, tree-, and graph-oriented. A main observation was that pattern-based methods were the first to be developed for RE, while tree-oriented approaches were extensively adapted between the 2000s and 2010. Later on, neural architectures initially utilised sequence-oriented techniques that performed on long or shorter sequences. These methods, however, were further augmented with tree-structured information or explicit transformation of the input into a tree structure, proving the effectiveness and popularity of such formations for the task of relation extraction. However, the flexibility of tree structures was limited, despite their efficacy. Graph-based methods emerged as a result, where initial efforts incorporated graphs for cross-level relation extraction. Graph-based methods are consistently used in latest approaches for RE since they provide more leeway in connecting various elements, thus capturing latent information in the input snippet. Their success has motivated our proposed approach of be graph-oriented, where we experimented with various types of graphs; either homogeneous, heterogeneous, partially or fully connected. Nonetheless, the search space in such methods can increase significantly and, as a result, a variety of pruning techniques has been introduced to remove non-informative connections.

It is important to note that it is fairly hard to strictly categorise prior work into too fine-grained categories, due to the diversity of methods and learning techniques explored over the years. Looking over the large variety of structure-oriented RE methods in the timeline, we concluded that each has its own benefits and drawbacks, which is why most approaches are actually hybrid in nature. Figure 2.9 illustrates the timeline of the aforementioned approaches, showing the underlying textual structure of each approach with different colours. We focus on showcasing the methods that had the most impact on the field over time, leaving more detailed work to be described in subsequent

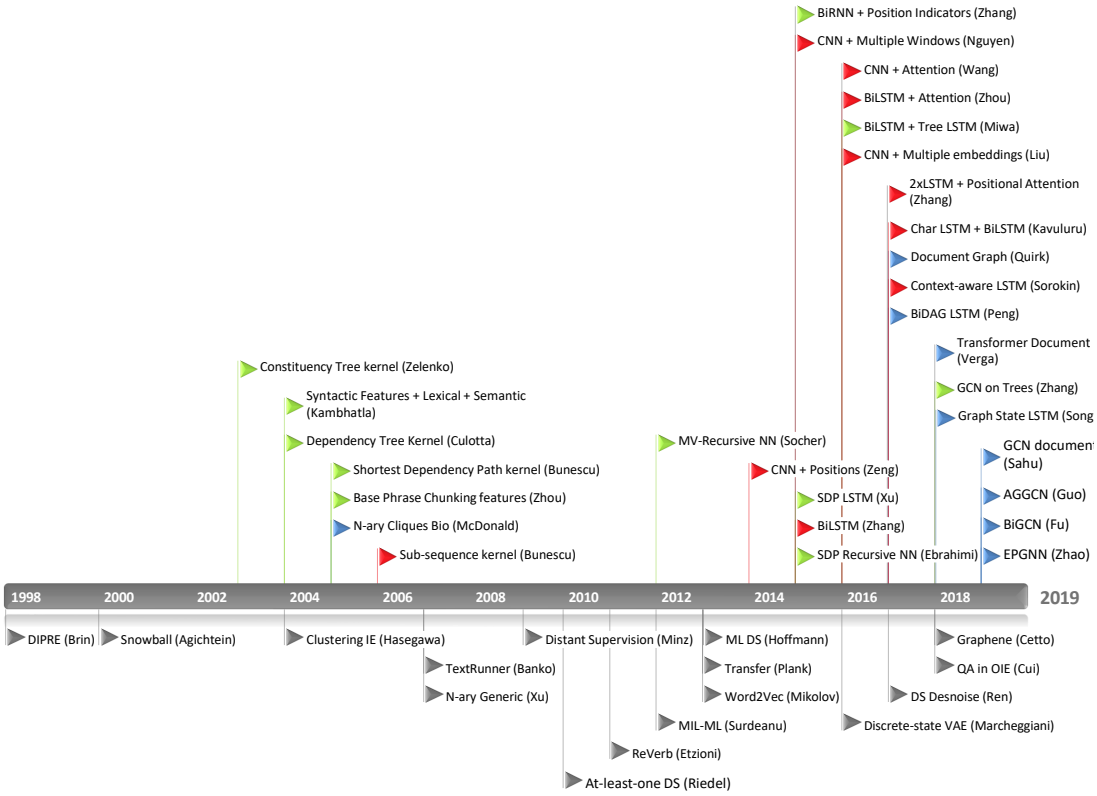


Figure 2.9: Timeline of existing methods for Relation Extraction. Red flags indicate sequence-oriented approaches, green flags indicate tree-oriented approaches and blue flags indicate graph-oriented approaches. The flags under the timeline correspond to non fully supervised methods.

chapters.

Despite the many approaches proposed for RE, there is still a number of limitations that we aim to tackle in this work. A first problem among existing approaches is that they treat multiple pairs in a certain context (particularly sentences) in isolation, thus failing to model higher interactions among pairs. Another limitation is that they do not model the directionality of relations, with the exception of approaches applied on the SemEval 2010 shared task, that explicitly required directed relations. In Chapter 4, we will discuss the fact that graph-based approaches can model directionality thus being useful in modelling directions, which is an important property of relations. Furthermore, we address the issue of portability of approaches in relation extraction which, in their majority, use external tools such as parsers, to enhance performance. A drawback of such approaches is that direct application to different domains is not straightforward, since the corresponding tools need to be adapted as well. We thus aim to propose a non-domain specific approach that can be effectively used for relation extraction in the

biomedical domain. It is important to mention, however, that a limitation of the methods that will be presented in future chapters, is that we consider named entities already extracted and annotated, thus focusing only on extraction of relations among them, for both sentence-level and document-level RE.

Chapter 3

Technical Background

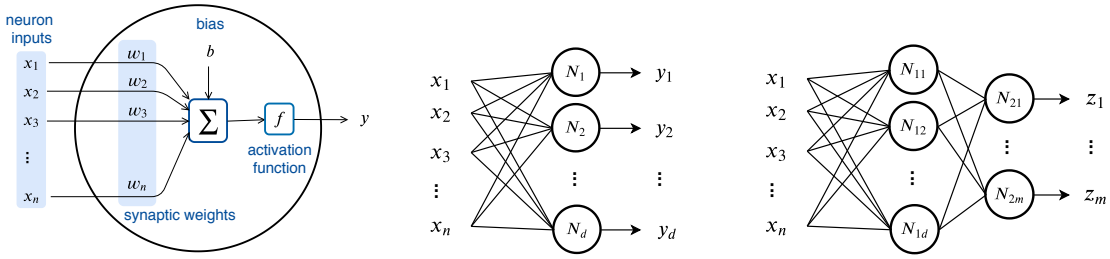
In this chapter we provide fundamental technical background for the remainder of this dissertation. We provide a short introduction to Artificial Neural Networks and present fundamental information about basic network architectures used in the following chapters, as well as information regarding their training procedure. In essence, this chapter serves as a brief introduction to neural networks in order to facilitate the reader in the following chapters.

3.1 Artificial Neural Networks

Artificial Neural Networks (ANN) or Neural Networks (NN), for short, are computing tools that were inspired by efforts to simulate the function of the brain. They can be seen as a graph, with nodes (neurons) and connections between them. These networks that were modelled by McCulloch and Pitts (1943), were initially implemented as electric circuits. Then, Hebb (1962) supported the concept of the artificial neuron, where he proposed *Hebbian learning*, a form of learning where the neural nodes strengthen their associations when used simultaneously. The computation performed in a single neuron, also known as a Perceptron, is the following: Each incoming connection to the neuron has an associated weight which is multiplied with its corresponding input. The output of the neuron is the weighted summation of its inputs (with an additional bias value), passed from an *activation function* that controls the amplitude of the output. The neuron computation can be seen visually in Figure 3.1a or in the form on an equation as,

$$y = f\left(\mathbf{w}^\top \mathbf{x} + b\right), \tag{3.1}$$

where y is the output of the neuron, b is a scalar bias weight, $\mathbf{x} \in \mathbb{R}^n$ are the network inputs and $\mathbf{w} \in \mathbb{R}^n$ is a vector with the synaptic weights.



(a) Perceptron. (b) Single-layer Perceptron. (c) Two-layer Perceptron.

By combining multiple neurons together, we can construct layered networks. The first of these networks is considered to be the single-layered Perceptron, whose algorithm was proposed by Rosenblatt (1957). We can imagine that the layered Perceptron consists of d neurons (N) that share the same input (Figure 3.1b). Analogously with the computation of a neuron, we can write the computation of a single-layered Perceptron as,

$$\mathbf{y} = f(\mathbf{W}\,\mathbf{x} + \mathbf{b}), \tag{3.2}$$

where f corresponds to an activation function, $\mathbf{W} \in \mathbb{R}^{d \times n}$ is a synaptic weight matrix multiplied by the input vector $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^d$ is a bias vector with dimensionality d and $\mathbf{y} \in \mathbb{R}^d$ is the output vector of the network.

The Perceptron is known as a binary classifier, able to solve linearly separable problems. The most famous example that shows the inability of the Perceptron to solve non-linearly separable problems is that of the boolean XOR operator. Minsky and Papert (1969) proposed the multi-layered Perceptron (MLP) for this purpose. The introduction of one additional (hidden) layer in the network enabled the projection of the data into a different space, where linear separability is feasible. Now the output of a two-layer network (as seen in Figure 3.1c) can be written as,

$$\mathbf{z} = f_2(\mathbf{W}_2\, f_1(\mathbf{W}_1\,\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2), \tag{3.3}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{z} \in \mathbb{R}^m$, $\mathbf{W}_1 \in \mathbb{R}^{d \times n}$, $\mathbf{W}_2 \in \mathbb{R}^{m \times d}$, $\mathbf{b}_1 \in \mathbb{R}^d$ and $\mathbf{b}_1 \in \mathbb{R}^m$. By stacking several layers, one can create very deep neural architectures. The single layer Perceptron is the most classic feed-forward neural network, in the sense that its computation does not form cycles.

Mapping elements to vectors

An elemental property of learning with neural networks is that they represent inputs, outputs and intermediate states using vector representations. The transformation of the input of a network into real-valued vectors is known as the *embedding layer*. For Natural Language Processing tasks and particularly text, the input is considered a sequence of words, i.e. the input is discrete. As such, converting words to vectors is necessary when working with neural networks. On the contrary, the inputs are not discrete in computer vision, i.e. images are treated as a matrices of pixels (each pixel has a numerical value) and are fed into the NN model directly. Hence, there is no need for an embedding layer. Since we aim to work with text, we introduce the traditional way to construct an embedding layer as follows.

Firstly, we define a dictionary D that contains all the words of the dataset that we want to use. Secondly, we define a mapping function in order to associate a word with a real-valued feature vector. For this purpose, a look-up table $\mathrm{LT}_W(\cdot)$ is constructed (Collobert and Weston, 2008), where each word $i \in D$ is embedded into a d-dimensional feature vector,

$$\mathrm{LT}_W(i) = \mathbf{W}_i, \tag{3.4}$$

where $\mathbf{W}_i \in \mathbb{R}^d$ is the i^{th} column of matrix W and d is the dimensionality of the column vector. Following the same procedure, one can map other elements into vectors, such as relative distances (i.e. signed integers), sub-words, etc.

3.2 Network Training

In this section we introduce and briefly describe techniques used for training neural networks. The methods that follow are primarily used as tools for the research conducted in this dissertation. We additionally provide fundamental information that can facilitate the understanding of their functionality.

3.2.1 Classification and Cost Function

In the beginning of Section 3.1, we defined the output of a multi-layer neural network as a vector $\mathbf{z}$. In classification problems, the last layer of the network typically maps the output to a vector of dimensionality equal to the number of potential classification categories. The values of this vector can be seen as un-normalised scores, one for each of these categories. If C is the number of classification categories, then the vector is

$\mathbf{z} \in \mathbb{R}^C$. In order to convert the un-normalised scores into probabilities, we employ the softmax activation function. This helps us compute the probability q_c of a category $c \in C$ given an input vector $\mathbf{x}$ as follows,

$$q_c = \text{softmax}(z_c) = \frac{\exp(z_c)}{\sum_{k=1}^{C} \exp(z_k)}, \tag{3.5}$$

where z_c is the un-normalised score as resulted from the network for category c. Softmax can be considered as a "soft argmax" operation. It amplifies the largest un-normalised score and normalises the scores of the output in order to reside in $[0,1]$ and sum to one.

Loss Function

In order to train NN models, we need to define a function that can help us measure, in a quantitative way, how satisfied we are with the predictions that our model produced. This function is referred to as *loss function*.

Let us assume we have a multi-class problem, with C being the number of classification categories, and a dataset with N examples $\{\mathbf{x}_i, c_i\}_{i=1}^{N}$, where $\mathbf{x}_i$ is the input vector corresponding to example i and c_i is the correct category associated with this example, which is an integer. Let $f(\mathbf{x};\theta)$ be an abstractly defined function that represents our NN model, with $\mathbf{x}$ our input examples and θ being the model's parameters (e.g. matrices). In general, the loss function over our entire dataset can be defined as the average of the losses of each example:

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} L_i \left(f(\mathbf{x}_i;\theta), c_i\right) \tag{3.6}$$

Our training objective is to minimise the loss with respect to θ. Formalising this objective, we have:

$$\hat{\theta} = \underset{\theta}{\text{argmin}}\ L(\theta) \tag{3.7}$$

There are several loss functions used to train neural networks. However, the most vastly used for classification problems is the cross-entropy loss. We assume that the output of the NN model for the i-th example is the probability distribution $\mathbf{q}_i$ over a set of categories, as resulted after the softmax activation. The true category distribution for each example can be expressed as a one-hot vector $\mathbf{p}_i$ with a probability equal to one for the correct category and a probability equal to zero for all the other categories,

such as $p_{i,c_i} = 1$ and $p_{i,c_j} = 0$, where p_{i,c_i} is the c_i-th element of $\mathbf{p}_i$. Since we have two distributions, we can measure their dissimilarity. This information can be used to estimate our satisfaction with the model predictions, hence forming a loss function.

Our goal is to encourage our model to produce a distribution that matches the correct distribution for each example. In order to realise that, we can use the Maximum Likelihood Estimation (MLE) principle to maximise the likelihood of predicting the correct category for each example. Translating this into a loss function (that we want to minimise), we need to maximise the log likelihood thus minimise the negative log likelihood of the correct category, for example i:

$$L_i(\theta) = -\log\left(f(\mathbf{x}_i;\theta), c_i\right) = -\log\left(q_{i,c_i}\right), \tag{3.8}$$

where, q_{i,c_i} is the probability of the NN model to predict the correct category c_i given example i with θ parameters (later omitted for readability). The reason that we work with the log-likelihood is because of the faster and easier computation that the log function results in computers by avoiding potential under-flows.

If we take the average of the losses over all the input examples we have,

$$\begin{aligned} L(\theta) &= \frac{1}{N}\sum_{i=1}^{N} L_i(\theta) = \frac{1}{N}\sum_{i=1}^{N} -\log\left(q_{i,c_i}\right) \\ &= \frac{1}{N}\sum_{i=1}^{N} -\left(\sum_{k=1}^{C} p_{i,k}\,\log(q_{i,k})\right) \\ &= \frac{1}{N}\sum_{i=1}^{N} H\left(\mathbf{p}_i, \mathbf{q}_i\right) \end{aligned} \tag{3.9}$$

The last term is the cross-entropy between the actual distribution $\mathbf{p}_i$ and the predicted distribution $\mathbf{q}_i$, since $p_{i,k}$ is equal to one only when $k = c_i$. Hence, the maximisation of the log-likelihood is the same as the minimisation of the cross-entropy for each example, which gave its name to the loss function. In binary classification problems, the loss function is referred to as *binary cross-entropy* loss, while in multi-class classification problems we also use the term *categorical cross-entropy* loss.

3.2.2 Learning

Back-propagation

In order for a network to be trained, we need to use the computed loss and update its parameters accordingly. This can be realised using the back-propagation technique. Back-propagation is the backbone of neural networks training. The back-propagation algorithm (Rumelhart et al., 1988) essentially provides an effective way to calculate and transfer the error of the model into its parameters throughout the entire network. This technique, or also known as *error propagation*, computes the gradient of a loss function, with respect to each parameter of the network.

In order to apply back-propagation, the loss function needs to be differentiable with respect to the parameters of the network. Since, in complex networks, the loss is dependent on multiple parameters, we compute the partial derivatives of the loss with respect to each parameter in a backward manner, i.e. from the loss to the parameters of the very first layer. For this purpose, we employ the gradient chain-rule. To briefly describe how back-propagation works, consider an example neural network f that does the following computation,

$$f(x,y,z) = x\,y + z, \;\; \text{if} \;\; a = x\,y, \;\; \text{then} \;\; f = a + z,$$

where we defined a as in intermediate result. For now consider that x, y, z are all scalars. We want to compute the gradient of the output with respect to the network inputs, as

$$\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$$

Starting from the network output, we can compute the gradient backwards as follows,

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial a}\,\frac{\partial a}{\partial x} = 1 \cdot y = y, \tag{3.10}$$

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial a}\,\frac{\partial a}{\partial y} = 1 \cdot x = x, \tag{3.11}$$

$$\frac{\partial f}{\partial z} = 1 \tag{3.12}$$

From this procedure we can see that the gradient of the intermediate result a helped us compute the gradient of the output f, which essentially constitutes the chain-rule. At each computation node (e.g. a or f) we can simply compute the local gradient that corresponds to the gradient of the output of the node, with respect to one of the inputs,

and multiply this with the upstream gradient, which is the gradient that comes from the next computation node, as shown in the schematic of Figure 3.2.

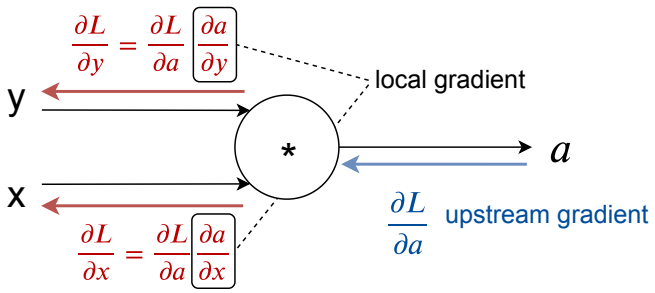


Figure 3.2: Abstract schematic of backward computation for a multiplication unit. Adaptation from cs231n.stanford.edu

In case we have vectors instead of scalars, the computation of the chain rule is the same, though now we compute the Jacobian matrix, which contains the derivatives of each element of the output with respect to each element of the input. The shape of the gradient with respect to a variable is the same as that of the variable. Let us suppose we have the following neural network, again using an intermediate output vector $\mathbf{a}$, that results in a scalar f:

$$f(\mathbf{x}, \mathbf{W}) = ||\mathbf{W}\,\mathbf{x}||^2, \;\; \text{if} \;\; \mathbf{a} = \mathbf{W}\,\mathbf{x}, \;\; \text{then} \;\; f = ||\mathbf{a}||^2$$

We want to compute the gradient of the output with respect to the input as,

$$\frac{\partial f}{\partial \mathbf{x}}, \frac{\partial f}{\partial \mathbf{W}}$$

Starting again from the output and moving backwards, we have,

$$\frac{\partial f}{\partial \mathbf{W}} = \frac{\partial f}{\partial \mathbf{a}}\frac{\partial \mathbf{a}}{\partial \mathbf{W}} = 2\mathbf{a}\,\mathbf{x}^\top, \tag{3.13}$$

$$\frac{\partial f}{\partial \mathbf{x}} = \frac{\partial f}{\partial \mathbf{a}}\frac{\partial \mathbf{a}}{\partial \mathbf{x}} = 2\mathbf{W}^\top\,\mathbf{a}\,, \tag{3.14}$$

This process is extended and applied to more complicated neural models with several layers. The main advantage of back-propagation lies in the fast computation of all the gradients throughout the entire network.

Parameter Optimisation

After estimating the partial derivatives of each parameter in the network, we need to update the network parameters θ in order to minimise the loss function $L(\theta)$ (Equation (3.7)). This is achieved via parameter optimisation. The formulation of the best parameters is achieved by iterative parameter updates during training.

One of the first optimisation methods was that of **Gradient Descend** (GD) (Cauchy, 1847), where the weights were updated globally on the entire training set, towards the opposite direction of the gradient of the loss L:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta L(\theta_t) \tag{3.15}$$

where η corresponds to the learning rate and t indicates the iteration step. In more detail, each parameter is only updated using a part of its partial derivative, which is controlled via the η hyper-parameter called the *learning rate*. Typically, large learning rates result in large changes to the network weights, while small learning rates result in small changes and consequently smoother learning.

The motivation behind GD was that we need to find the minimum point of our loss, hence we need to traverse the parameter space by going towards the direction that will lead to the lowest point of the loss. This point can be found moving along the negative direction of the gradient, that will lead us to the steepest descent (opposite from the steepest ascent which is represented by the (positive) direction of the gradient). However, if our training set is very large, it is very slow and expensive to compute the gradient of the loss for the entire dataset before we make an update. Therefore, in practice, we use **Stochastic Gradient Descent** (SGD), which aims to update the network parameters for a randomly chosen set of training examples, named a *mini-batch*. For each mini-batch of $n < N$ examples, an estimate of the true gradient is computed over the selected examples and an update is made based on these examples only.

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \frac{1}{n} \sum_{i=1}^{n} L_i(\theta_t) \tag{3.16}$$

SGD suffers from several problems, the most important of which being that it has trouble navigating along curves that are steep in one direction but not so steep in another (saddle points). Additionally, SGD updates the model parameters at each step using a noisy estimation of the actual gradient, so it might take a very long time to converge. In order to avoid these problems, **SGD with momentum** (Qian, 1999) was

proposed. The idea was based on updating the parameters towards the direction of a velocity (moving average, or simply average of gradients) rather than the gradient itself, using a parameter ρ that can be seen as corresponding to friction,

$$v_{t+1} = \rho\, v_t + \eta \nabla_\theta \frac{1}{n} \sum_{i=1}^{n} L_i(\theta_t), \tag{3.17}$$

$$\theta_{t+1} = \theta_t - v_{t+1} \tag{3.18}$$

This way, velocity is accumulated through time and the parameters are updated faster or slower, depending on the direction of the gradients.

Another widely used optimisation algorithm is **Adam** (Kingma and Ba, 2015), which estimates adaptive learning rates for each parameter θ of the network. Adam combines SGD's momentum (average of gradients), along with an exponentially decaying average of squared gradients similar to AdaDelta (Zeiler, 2012).

$$m_{t+1} = \beta_1\, m_t + (1-\beta_1)\, g_{t+1}, \qquad \hat{m}_{t+1} = \frac{m_{t+1}}{1-\beta_1^{t+1}}, \tag{3.19}$$

$$u_{t+1} = \beta_2\, u_t + (1-\beta_2)\, g_{t+1}^2, \qquad \hat{u}_{t+1} = \frac{u_{t+1}}{1-\beta_2^{t+1}}, \tag{3.20}$$

$$\theta_{t+1} = \theta_t - \eta\, \frac{\hat{m}_{t+1}}{\sqrt{\hat{u}_{t+1}} + \varepsilon} \tag{3.21}$$

where $g_{t+1} = \nabla_\theta\ L\big(\theta_{t+1}\big)$ are the network weights gradients. Other optimisation approaches include RMSProp (unpublished) and AdaGrad (Duchi et al., 2011).

Regularisation

During training, a neural model is typically evaluated on another set of data, named the *validation set* (also known as *development set*). In general, we aim at creating models with good generalisation ability, i.e., models that are able to make decent predictions, not only on the validation set but also on another, blind data set known as the *test set*. However, models are trained on the training set and the loss function is minimised on this set. As a consequence, if the NN model is expressive enough, it can *overfit* on the training data. The opposite scenario involves what is known as *underfitting*, i.e. the model is unable to fit the training data. In the former case, the model results in higher generalisation error when applied on unseen data. Regularisation techniques have thus been developed in order to mitigate this problem. We briefly discuss some of the most well-known regularisation techniques used in neural networks.

Weight Decay. This type of regularisation is the one most commonly used to avoid over-fitting. It essentially penalises the weights of a network by a controllable scalar λ, as follows,

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} L(\theta) + \lambda R(\theta) \tag{3.22}$$

The parameter update will take place as,

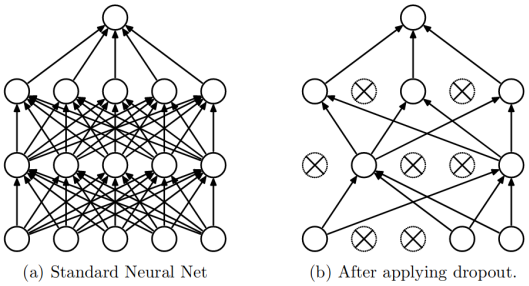
$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \ L(\theta_t) - \eta \lambda \nabla_\theta R(\theta), \tag{3.23}$$

Two common weight decay regularisers are the L_1 and L_2 regularisation, described in Equations (3.24) and (3.25). Since the gradient of L1 has two possible outcomes ($+1$ and -1), in the first case 1 will be subtracted from θ while, in the second, 1 will be added. In both cases θ tends to be pushed towards zero. Hence, L1 prefers more sparse parameters for multi-dimensional problems leading to fewer variables and simpler models.

$$R_{L_1}(\theta) = \|\theta\|_1 = \sum_i |\theta_i|, \tag{3.24}$$

$$R_{L_2}(\theta) = \|\theta\|_2^2 = \sum_i \theta_i^2 \tag{3.25}$$

On the contrary, the derivative of L2 substracts a portion of 2θ from the θ parameter. The penalisation is proportional to the θ parameter. As a result, L2 regularisation forces all network parameters to be relatively small, though not exactly zero, leading to non-sparse parameters.



(a) Standard Neural Net (b) After applying dropout.

Figure 3.3: Dropout illustration (Srivastava et al., 2014).

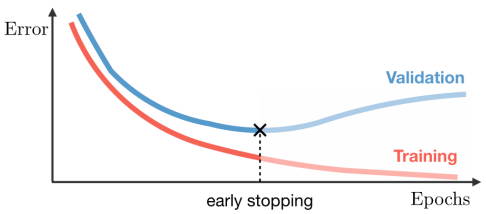


Figure 3.4: Early stopping criterion. Source: stanford.edu

Dropout. Dropout (Srivastava et al., 2014) was proposed as a method to prevent neural models from relying on specific weights. In essence, the technique randomly replaces

weights in the network with zeros, i.e. "dropping" units (Figure 3.3). Dropout randomly drops units at each iteration step, and can thus be considered as generating a different model each time, finally producing an ensemble. This is one of the reasons for the technique's success, while the other is considered to be the incorporation of noise into the network through the missing units. Dropout makes the training slower in general, but enables models to generalise better. Last, but not least, the probability of dropping units can be tuned for different layers: input, hidden or output ones.

Early Stopping. Early stopping is another regularisation technique used in practice while training neural models (Caruana et al., 2001). It is essentially a heuristic that assumes that, if the loss of a model on the validation set starts to increase but the training loss continues to decrease, training must stop due to over-fitting (Figure 3.4). Typically, early stopping involves choosing a hyper-parameter, named *patience*. The value of patience indicates the number of training epochs that we wait for, after observing the validation loss increasing (and the training loss decreasing). If, after a number of continuous epochs (*patience*), the validation loss keeps decreasing, we stop the training and choose the model parameters that resulted in the best (lowest) validation loss.

3.3 Neural Components

Given the necessary information for network training, we describe the most important neural networks that are used in NLP applications.

3.3.1 Convolutional Neural Networks

A family of feed-forward neural networks, named Convolutional Neural Networks (CNN), were initially introduced for image processing (Sharma et al., 2018). Their name stems from their core computation, convolution, a linear, mathematical operation that can replace matrix multiplication (Smith et al., 1997). The origins of CNNs lay in neuroscience. Their inspiration was based on an experiment conducted by Hubel and Wiesel (1962), where they observed that some neurons in the visual cortex of the brain of cats were triggered by the presence of specific edges of certain orientation. For instance, some neurons responded to vertical, horizontal or diagonal edges. The researchers additionally observed that these neurons were organised in column-ordering

inside the brain, producing visual perception. This observation resulted in the theoretical basis of CNNs, where different neural components exist and are responsible for encoding different types of information.

Vanilla CNN

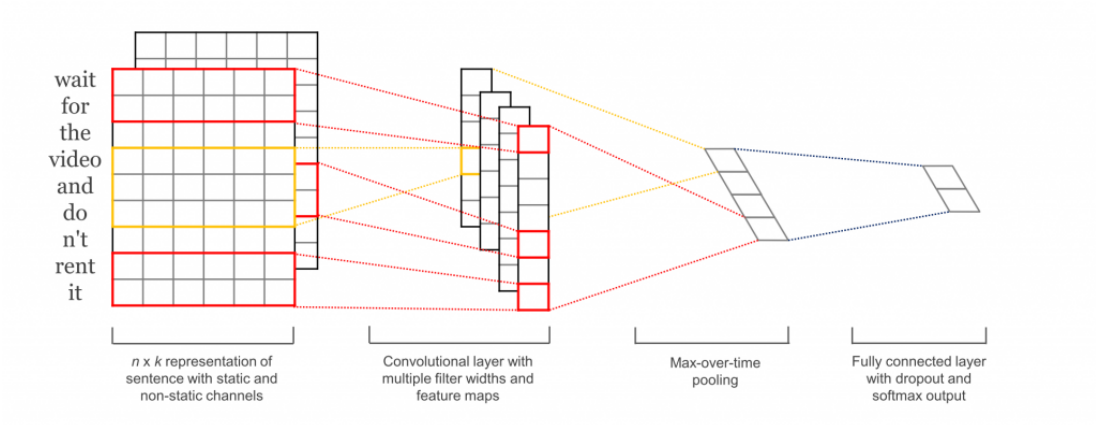


Figure 3.5: Architecture of a Convolutional Neural Network (CNN) (Kim, 2014).

CNNs were firstly applied on images, with LeCun et al. (1989) proposing automatic training of the convolution kernel. These networks have already gained significant interest in the domain of image processing, where they still constitute the core computational network of the domain. Their introduction to text came much later, with one of the first approaches presented in Kim (2014). Their basic computation on sequences of words for sentence representation is depicted in Figure 3.5. The input is a sentence, which is converted into a numerical matrix, with each word being associated with a real-valued vector (embedding layer). The architecture of a CNN network consists of three building blocks which, as a group, can be repeated in order to form deeper networks. These blocks are: a convolution layer, a non-linear layer and a pooling layer.

Consider an input word represented as a k-dimensional vector $\mathbf{x}_i \in \mathbb{R}^k$. A sentence is then represented as the concatenation of n words as $s = \mathbf{x}_{1:n} = [\mathbf{x}_1; \mathbf{x}_2; , \cdots , ; \mathbf{x}_n]$, with ";" denoting the concatenation operation. In order to perform convolution across the sentence words, we use a *filter* $\mathbf{f} \in \mathbb{R}^{hk}$, with dimensionality equal to the product of the number of sequential words h and their dimension k. The number of sequential words is also known as a *window* of size h. The convolution operation with one filter on a

window of h words produces a single feature c_i, as shown in Equation (3.26):

$$c_i = g\left(\mathbf{f}^\top \mathbf{x}_{i:i+h-1} + b\right) \tag{3.26}$$

where b is a scalar, $\mathbf{x}_{i:i+h-1} \in \mathbb{R}^{hk}$ is the concatenation of h words and g is a non-linear activation function. The convolution operation is applied on all possible slices of the sentence $\{\mathbf{x}_{1:h}, \mathbf{x}_{2:h+1}, \ldots, \mathbf{x}_{n-h+1:n}\}$ creating a set of features, named the *feature map*, Equation (3.27).

$$\mathbf{c} = [c_1, c_2, \cdots, c_{n-h+1}], \tag{3.27}$$

$$\hat{c} = \max(\mathbf{c}) \tag{3.28}$$

A max pooling operation, Equation (3.28), is then applied on the feature map $\mathbf{c}$ to extract the most prominent feature $\hat{c}$ for this filter $\mathbf{f}$. CNNs typically use multiple filters, with different window sizes, to extract as many feature maps as possible. After the pooling operation of each one of m filters, the resulted features are concatenated into a single vector $\mathbf{v} = [\hat{c}_1; \hat{c}_2; , \cdots, ; \hat{c}_m]$ and fed into a fully connected layer.

Essentially, CNNs are able to extract n-grams through different filter sizes. As a result, they can perform n-gram composition, i.e. words that consist of up to n characters or phrases with n words. The max pooling operation ensures that only the best-matched n-gram will be selected out of the applications of a specific filter to continuous windows of the input.

Graph CNN

A recently developed neural model based on CNNs is the Graph CNN (GCN), proposed for application on graph structures instead of sequences (Kipf and Welling, 2017; Marcheggiani and Titov, 2017). GCNs aim to learn an informative representation for each node in the graph, taking as input a set of nodes with corresponding representations and an adjacency matrix. Their key operation is to take advantage of neighbouring node information and update the current node representation iteratively. As shown in Figure 3.6, for a simple non-directed graph during the first iteration, a node (marked in grey) is updated by aggregating information from its immediate neighbours (marked in blue). The process is repeated in the second iteration, although this time, the node of interest (grey) is updated with information included in further nodes (2-hops away) since its immediate neighbours included this information in the

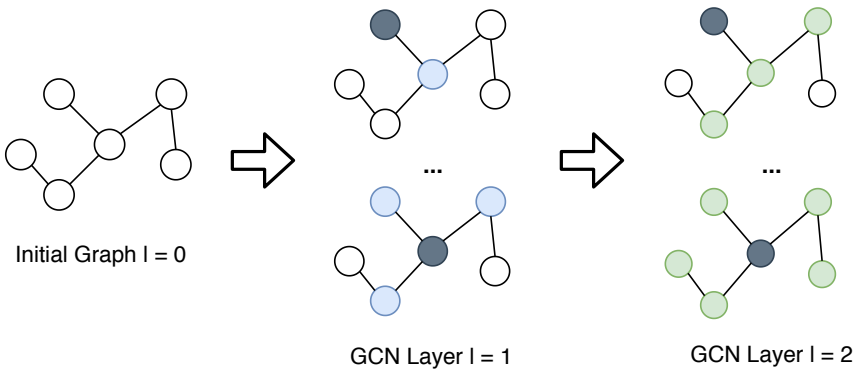


Figure 3.6: Abstract representation of Graph Convolutional Neural Networks.

previous step.

In order to achieve that, we stack multiple GCN layers on top of one other to encode information from distant neighbouring nodes in the current node,

$$\mathbf{v}_i^{(l+1)} = f\left(\sum_{u \in v(i)} \mathbf{W}^{(l)}\, \mathbf{v}_u^{(l)} + \mathbf{b}^{(l)}\right), \tag{3.29}$$

where $\mathbf{v}_i^{(l+1)} \in \mathbb{R}^m$ is the representation of the node of interest resulted from the l-th GCN layer, $u \in v(i)$ is a set of neighbouring nodes to v_i, $\mathbf{v}_u \in \mathbb{R}^d$ is the representation of a neighbouring node, $\mathbf{W}^{(l)} \in \mathbb{R}^{m \times d}$ and $\mathbf{b}^{(l)} \in \mathbb{R}^m$ are learnable parameters of the l-th GCN hidden layer and f as a non-linear function. GCNs can be found in different variations than the one described in Equation (3.29). For instance, instead of summing the neighbouring node representations, we can normalise the adjacency matrix in order to estimate the final node representation as the average of the representations of the neighbouring nodes.

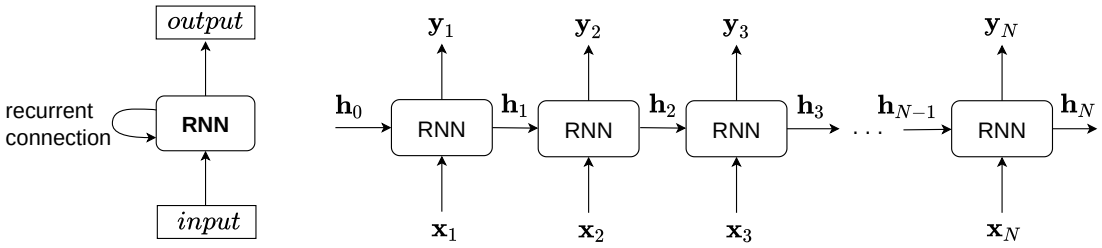
3.3.2 Recurrent Neural Networks

In comparison with CNNs and Perceptron, Recurrent Neural Networks (RNNs) are a special type of network that was developed to be applied on sequences. In particular, they are ideal for sequence-to-sequence problems such as Machine Translation (Cho et al., 2014), where the input is a sequence of arbitrary length and the output is another sequence of possibly different length. There are two main versions of RNNs; the vanilla RNNs, which constitute the base model, and Long-Short Term Memory Networks (LSTM), a different variant that overcomes certain problems of the vanilla version. There are more RNN-based models as well, such as the Gated Recurrent Unit

(GRU). However, we do not include them here as they are not used in the models proposed in this thesis.

Vanilla RNN

Recurrent Neural Networks (RNN) were originally introduced by Elman (1990). These networks rely on a special connection from themselves to themselves, called the *recurrent connection*. This enables them to be applied on sequences, such as words in a sentence. The characteristic of RNNs is that a single RNN cell can be unfolded in order to perform the same operation on each element of a sequence, using the previous element as additional input. This modelling has been referred to as *memory*, from the perspective that RNNs are able to *remember* information about their previous inputs. This memory is represented in the form of a vector, also known as the *hidden state*.



(a) Single RNN cell. (b) Repeating module of a vanilla RNN.

Figure 3.7: Architecture of the Recurrent Neural Network (RNN).

As illustrated in Figure 3.7a, the RNN cell receives input and produces output. However, if we unfold the recurrent connection (Figure 3.7b), we can imagine each piece of input in a sequence passed through an RNN cell along with the previous hidden state. The initial hidden state is typically initialised with zeros. The computation that takes place inside each cell is described in Equations (3.30)-(3.31), as a two-layer neural network.

$$\mathbf{h}_t = \tanh\left(\mathbf{W}_{hh}\,\mathbf{h}_{t-1} + \mathbf{W}_{xh}\,\mathbf{x}_t + \mathbf{b}_{hh}\right), \tag{3.30}$$

$$\mathbf{y}_t = \mathbf{W}_{yh}\,\mathbf{h}_t + \mathbf{b}_{yh}, \tag{3.31}$$

where $\mathbf{W}_{xh} \in \mathbb{R}^{m\times d}$, $\mathbf{W}_{hh} \in \mathbb{R}^{m\times m}$ and $\mathbf{W}_y \in \mathbb{R}^{k\times m}$ are weight matrices, associated with the input, hidden and output vectors respectively, $\mathbf{b}_{hh} \in \mathbb{R}^m$ and $\mathbf{b}_y \in \mathbb{R}^k$ are bias vectors, $\mathbf{x}_t \in \mathbb{R}^d$ is the input vector at step t, $\mathbf{h}_t \in \mathbb{R}^m$ is the hidden state representation at step t, tanh is the hyperbolic tangent non-linear function and $\mathbf{y}_t \in \mathbb{R}^k$ is the output of

the RNN block at step t.

The weight matrices in RNNs are shared across all steps. RNNs are powerful networks that can be considered equal to Turing machines (Hyotyniemi, 1996). However, they suffer from the inability to model long term dependencies that arise from the problem of the vanishing gradient.

Long-Short Term Memory

The LSTM network was initially proposed by Hochreiter and Schmidhuber (1997) and its main advantage over the vanilla RNN is the ability to solve the problem of the vanishing gradient leading to effective encoding of longer sequences. In vanilla RNNs, in order to compute the gradient of the loss with respect to the hidden state, we need to traverse the entire sequence through back-propagation (Bengio et al., 1994; Pascanu et al., 2013). Since in a multiplication operation, e.g. $\mathbf{W} \cdot \mathbf{h}$, the gradient with respect to $\mathbf{h}$ is $\mathbf{W}^\top$, the gradient of the loss with respect to the initial hidden state $\mathbf{h}_0$ will be the power of the matrix to the length of the sequence. If the largest singular value of the weight matrix is larger than one then the gradient of the loss with respect to $\mathbf{h}_0$ will explode, which is known as an *exploding gradient*, while if the value is less than one the gradient will vanish, known as a *vanishing gradient*. Exploding gradients can be avoided with *gradient clipping*, by restricting the norm of the gradient in a specified range. On the other hand, vanishing gradients can be resolved with LSTMs.

LSTM is a specialised version of the RNN that is able to learn long-term dependencies. There are five hidden layers inside each LSTM cell that interact with each other. Four of these layers are named *gates* and are responsible for different computations in the cell. The last and most important layer is the *cell state*, which can be viewed as an artificial memory and is kept internally inside the cell. The following equations describe the function of each gate ($\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$, $\mathbf{g}_t$) the computation of the cell state ($\mathbf{c}_t$) and the LSTM output ($\mathbf{h}_t$) at east step t:

$$\mathbf{i}_t = \sigma\left(\mathbf{W}_{xi}\,\mathbf{x}_t + \mathbf{b}_{xi} + \mathbf{W}_{hi}\,\mathbf{h}_{t-1} + \mathbf{b}_{hi}\right), \tag{3.32}$$

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_{xf}\,\mathbf{x}_t + \mathbf{b}_{xf} + \mathbf{W}_{hf}\,\mathbf{h}_{t-1} + \mathbf{b}_{hf}\right), \tag{3.33}$$

$$\mathbf{o}_t = \sigma\left(\mathbf{W}_{xo}\,\mathbf{x}_t + \mathbf{b}_{xo} + \mathbf{W}_{ho}\,\mathbf{h}_{t-1} + \mathbf{b}_{ho}\right), \tag{3.34}$$

$$\mathbf{g}_t = \tanh\left(\mathbf{W}_{xg}\,\mathbf{x}_t + \mathbf{b}_{xg} + \mathbf{W}_{hg}\,\mathbf{h}_{t-1} + \mathbf{b}_{hg}\right), \tag{3.35}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t, \tag{3.36}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh\left(\mathbf{c}_t\right) \tag{3.37}$$

where $\odot$ denotes element-wise multiplication, $\mathbf{W}$ and $\mathbf{b}$ correspond to weight matrices and bias vectors, respectively.

Intuitively, the function of each gate controls the memory of the LSTM cell. Therefore, if we imagine an LSTM cell as our memory, we control how much we want to forget from our current cell state through the forget gate (Equation (3.33)). Incoming input information (represented as the output of Equation (3.35)) is controlled by the input gate (Equation (3.32)) before being added to the cell state. Finally, the amount of current memory ($\mathbf{c}_t$) that is revealed to the next hidden state is controlled by the output gate (Equation (3.34)). As described in Figure 3.8, our current memory $\mathbf{c}_t$ is influenced

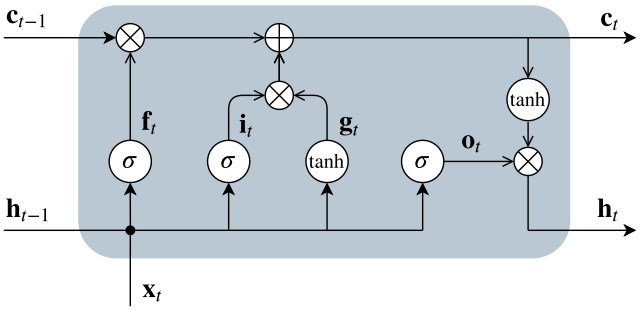


Figure 3.8: Abstract representation of an LSTM cell. The weight matrices $\mathbf{W}$ and bias vectors $\mathbf{b}$ are removed for brevity. The circled ($\times$) represents element-wise multiplication, while the circled plus (+) represents addition.

by the existing memory ($\mathbf{c}_{t-1}$) and how much we forget ($\mathbf{f}_t$) in addition to a portion of new information.

Bi-directionality

Bi-directionality was first introduced for RNNs by Schuster and Paliwal (1997) as an extension of vanilla RNN. The model was designed to utilise input from both past and future states at the same time, essentially increasing the amount of context information available to the network. The introduction of the opposite direction was particularly important when applied to LSTM variants (Graves and Schmidhuber, 2005; Thireou and Reczko, 2007). In practice, bi-directionality is applied using two distinct neural models applied from left-to-right and right-to-left on the input sequence. Typically, the outputs of the two networks are concatenated to form a final representation for each sequence output, as follows,

$$\mathbf{y}_t = [\overrightarrow{\mathbf{o}_t}; \overleftarrow{\mathbf{o}_t}], \tag{3.38}$$

where $\overrightarrow{\mathbf{o}_t}$ and $\overleftarrow{\mathbf{o}_t}$ correspond to the output of the left-to-right and right-to-left network, respectively; $\mathbf{y}_t$ is the final output of the bi-directional network and ";" indicates a concatenation operation.

3.3.3 Attention Mechanisms

The previously described models were heavily used for sequence-to-sequence tasks thanks to their sequential nature. Typically, two sequence-based networks were employed. An encoder that encodes the entire sequence into a single vector representation, and a decoder that generates (decodes) a new sequence from the given vector. The encoder-decoder architectures have been the core principle in machine translation. However, even though LSTMs can perform better when given longer sequences, it was observed that, for these particular tasks, some information is forgotten. A case of that is parenthetical text, typically inserted in the middle of a sentence. For this reason, it was necessary to find a way to pay more *attention* to particular words or parts of an input sequence that could play an important role in the production of the decoded sequence.

Attention mechanisms were initially developed for machine translation Bahdanau et al. (2015) in order to help the model remember important words that were located far away from the end of the sequence, or with respect to certain aspects. Several attention mechanisms were developed over the years, each one of them serving different purposes. However, all of these mechanisms try to measure the importance of different elements in a sequence with regard to other elements. In general, an attention mechanism can be defined as a function f that estimates a weight a_i for each element of a sequence $s = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, given input query $\mathbf{q}$. The attention weight is then normalised by the weights over the entire sequence.

$$\mathrm{a}_i = f(\mathbf{q}, \mathbf{x}_i), \tag{3.39}$$

$$\alpha_i = \mathrm{softmax}(\mathrm{a}_i) = \frac{\exp(\mathrm{a}_i)}{\sum_{j=1}^{N} \exp(\mathrm{a}_j)}, \tag{3.40}$$

where N is the length of the sequence and α_i is the normalised attention score associated with the element i. One of the benefits of attention mechanisms is that the attention scores can be easily visualised, leading to potential interpretability (Wiegreffe and Pinter, 2019).

Additive Attention. One of the first attention mechanisms was *additive attention* (also referred to as *concat attention* in Luong et al. (2015)), proposed by Bahdanau et al. (2015) for machine translation. The main idea behind it was to align the words of the input sentence (in the source language) with the words of another sentence (in the target language). Attention weights were estimated for each word representation $\mathbf{s}_t$ in the source sequence, with respect to the representation of a word $\mathbf{h}_i$ in the target sequence.

$$\mathrm{a}_{\text{additive}}(\mathbf{s}_t, \mathbf{h}_i) = \mathbf{v}_a^\top \tanh(\mathbf{W}_a[\mathbf{s}_t; \mathbf{h}_i]), \tag{3.41}$$

where $\mathrm{a}_{\text{additive}}$ is the attention score of the word pair (s_t, h_i), d is the dimensionality of the vectors that represent the words $\mathbf{s}_t$ and $\mathbf{h}_i$, ";" corresponds to the concatenation operation, $\mathbf{v}_a \in \mathbb{R}^m$ and $\mathbf{W}_a \in \mathbb{R}^{m \times 2d}$ are learned parameters of the attention network.

A simplified version of this type of attention was proposed by Zhou et al. (2016b) which learns the importance of a word i in a sequence $s = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ using a learned attention vector,

$$\mathrm{a}_{\text{vector},i} = \mathbf{u}_a^\top \tanh(\mathbf{x}_i), \tag{3.42}$$

where $\mathbf{u}_a \in \mathbb{R}^d$ is the learned attention vector, $\mathbf{x}_i \in \mathbb{R}^d$ corresponds to the vector of word i in the input sequence and $\mathrm{a}_{\text{vector},i}$ is the attention score for word i. Then, the final sequence representation is formed as the weighted average of the representations of its words.

Dot-Product Attention. Another type of attention was proposed by Luong et al. (2015), again for machine translation. In their approach, a direct multiplication was realised between the current word representation $\mathbf{s}$ in the source sequence and a word representation $\mathbf{h}$ in the target sequence. The inner product of the two vectors produces a scalar, which is the attention weight for the word t. A variation of the dot-product attention is that of *scaled* dot-product attention, introduced by Vaswani et al. (2017). Here, the attention weight is scaled by a scalar correlated with the dimensionality d of the representations, in order to avoid very small gradients due to large dimensionalities.

$$\mathrm{a}_{\text{dot}}(\mathbf{s}_t, \mathbf{h}_i) = \mathbf{s}_t^\top \mathbf{h}_i, \tag{3.43}$$

$$\mathrm{a}_{\text{scale-dot}}(\mathbf{s}_t, \mathbf{h}_i) = \frac{\mathbf{s}_t^\top \mathbf{h}_i}{\sqrt{d}}, \tag{3.44}$$

where $\mathbf{s}_t$ and $\mathbf{h}_i \in \mathbb{R}^d$.

Self-attention. However, even without paying attention to specific words of a sequence, we can allow the sequence to attend to oneself highlighting important aspects of it (words or phrases) using *self-attention*. Self-attention was proposed by Lin et al. (2017) and applied to several NLP tasks, including not only relation extraction, but also sentiment analysis (Montoyo et al., 2012) and textual entailment (Dagan and Glickman, 2004). The attention vector a_i for a word i is estimated in a similar manner to additive attention,

$$\mathrm{a}_i = \mathbf{v}_a^\top \tanh\left(\mathbf{W}_a \, \mathbf{h}_i\right), \tag{3.45}$$

where $\mathbf{W}_a \in \mathbb{R}^{m \times d}$, $\mathbf{v}_a \in \mathbb{R}^m$ are learned attention parameters and $\mathbf{h}_i \in \mathbb{R}^d$ corresponds to the representation of word i in the input sequence. Self-attention, however, is applied several times on the input sequence to extract multiple important spans. In this case, the vector $\mathbf{v}_a$ becomes a matrix $\mathbf{V}_a \in \mathbb{R}^{m \times r}$.

$$\mathbf{a}_i = \mathbf{V}_a^\top \tanh\left(\mathbf{W}_a \, \mathbf{h}_i\right), \tag{3.46}$$

where $\mathbf{a}_i \in \mathbb{R}^r$ is a vector with r attention weights for word i of the input sequence. The authors further added a penalisation term to the attention parameters in order to force the model to attend to different sections of the sequence.

Multi-head Attention. Multi-head attention is an extended adaptation of the dot-product attention and self-attention proposed by Vaswani et al. (2017). This type of attention is the core component of the Transformer network, a powerful network based on attention that surpasses the performance of the previously described sequential encoders (RNN and CNN) on various tasks. Based on this mechanism, several recent networks have been introduced, achieving state-of-the-art performance on several tasks (Devlin et al., 2019; Yang et al., 2019). We do not explain in more detail this particular type of attention, as we do not use it in the following parts of the thesis. We encourage readers to advise the recent literature for more details.

Chapter 4

Sentence-level Neural Relation Extraction

In this chapter, we introduce a novel neural model for sentence-level relation extraction on the generic domain, particularly newswire and Wikipedia. We address our first hypothesis (H_1), as introduced in Chapter 1: "The relation between two named entities in a sentence can be supported by the interactions of these entities with other, co-existing named entities in the same sentence, in a joint training setting". We propose a model that deviates from existing RE approaches, in that we treat all pairs in a sentence simultaneously and model their interactions with fixed dimensionality vector representations. We present an iterative algorithm that encodes interactions of named entity pairs in a sentence using all possible connections between the target entities. We evaluate our model on three multi-entity datasets, showing improvements over state-of-the-art methods. Additional analysis of the model's components substantiates important conclusions, not only with respect to the model's behaviour, but also to the characteristics of different relations between named entities. Parts of this work have been published in Christopoulou et al. (2018).

4.1 Motivation

As shown by previous work, the majority of developed approaches targeted the task of sentence-level RE. Most recent methods rely on deep neural networks, given their ability to efficiently encode important information into fixed-length vector representations. In this work, we also utilise deep neural network architectures based on their success in existing RE tasks (Miwa and Bansal, 2016; Nguyen and Grishman, 2015;

Zhang et al., 2018a, 2017b).

The aforementioned methods assume that a single pair resides in a sentence, thus ignoring co-existing pairs. As a result, the model encodes the representation of each pair separately, without considering latent interactions with other pairs. Despite this assumption, in real-life scenarios, a sentence typically contains multiple named entities and, consequently, multiple interactions among them. Moreover, it is expected that, at least some, if not all, of these pairs share commonalities with each other and their co-occurrence can further provide important information about their relation categories. This assumption was adopted by Sorokin and Gurevych (2017), where they constructed the representation of a pair of interest by incorporating the representations of other sentential pairs. They mostly address how co-occurrence of different relation categories can be beneficial in RE (i.e. *directed_by* often co-occurs with *produced_by*) rather than other semantic knowledge, such as compositionality. The proposed approach, however, has two limitations. The first is that it encodes the representations of pairs separately from one another in a sentence and then combines them. Secondly, it does not model the directionality of relations, hence using the direction of the pairs as given by the dataset.

Towards a similar direction, we try to simultaneously model multiple pairs in a sentence and take advantage of their interactions. To this end, we develop the following hypothesis:

> The relation between two named entities in a sentence can be *supported* by the *interactions* of these entities with other, co-existing named entities in the same sentence.

In order to better explain this intuition, one can look at the example illustrated in Figure 4.1, where the relation between a pair of interest (namely a "target" pair) can be influenced by other interactions in the same sentence. The relation between *troops* and *Basra* can be extracted in two ways: *directly*, by looking at the target entities and the sentence context, or *indirectly* by incorporating information from other pairs in the sentence, essentially breaking down the semantics of a sentence into smaller parts. The context around a pair can help us identify its relation. For instance, we can tell that the person entity (PER) *troops* is related to the location entity (LOC) *Iraq*, through preposition *in*. Similarly, *Basra* is related to the geopolitical entity (GPE) *Iraq*, with the preposition *near* as evidence. However, for the connection between *troops* and *Basra* we need to consider a more implicit connection, via the intermediate entity *Iraq*. In essence, the path from *troops* to *Iraq* to *Basra* can further support the relation between

troops and *Basra*, in addition to the existing context. Since *Basra* is part of *Iraq* and *troops* are *located in* (physically) *Iraq*, it is expected that they will share the same relation with *Basra*.

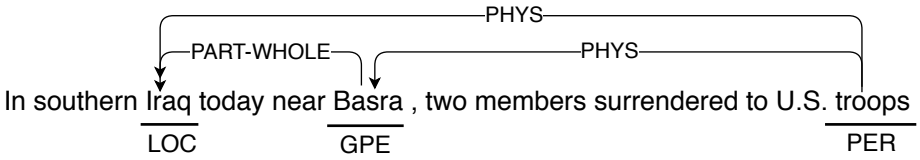


Figure 4.1: Relation examples from ACE (Automatic Content Extraction) 2005 dataset (Doddington et al., 2004).

Since we want to take advantage of multiple entities in a certain snippet and model their interactions, we do so via a graph structure. A graph can essentially be used to represent all named entity pairs alongside their interactions in a sentence simultaneously. Most state-of-the-art RE models depend on external syntactic tools to build graph or tree structures, such as the shortest dependency path (SDP) between two entities in a sentence (Xu et al., 2015a; Miwa and Bansal, 2016). Despite the offers of syntactic structures for relation extraction, their usage has two drawbacks. The first is that, when relations are implicit, the parser cannot return an informative path for the relation. The second is the dependence on external that leads to domain-restricted models. To tackle the above shortcomings, we build our model to be independent of domain-specific tools. In particular, we choose to place the named entities of a sentence as nodes in the graph and consider the relations between every two nodes as edges. We generate a unique representation for each edge and construct representations of entity chains iteratively, in order to model complex associations. We show that the proposed model can perform comparably well with models that use syntactic parsers not only for long-distance pairs, but also on sentences that include two or more entities. In the following sections, we first introduce the task at hand and then describe the proposed model in detail.

Task Definition

The particular Relation Extraction task that we target can be formally described as follows: Given a sentence s with words $\{w_1, w_2, \ldots, w_n\} \in s$ and identified named entities $\{e_1, e_2, \ldots, e_m\} \in E$, along with their semantic types $\{t_1, t_2, \ldots, t_m\} \in T$, the task is to identify a relation r from a set of pre-defined semantic relation categories R for each pair of named entities (e_i, e_j) in sentence s. The model's input is a sentence,

the named entities and their semantic types, while the output is a set of ordered triples in the form of (e_i, r, e_j) denoting that entity e_i has a relation r with entity e_j.

4.2 Proposed Approach

The proposed model architecture is illustrated in Figure 4.2. The input to the model is a sentence and the contained named entities.

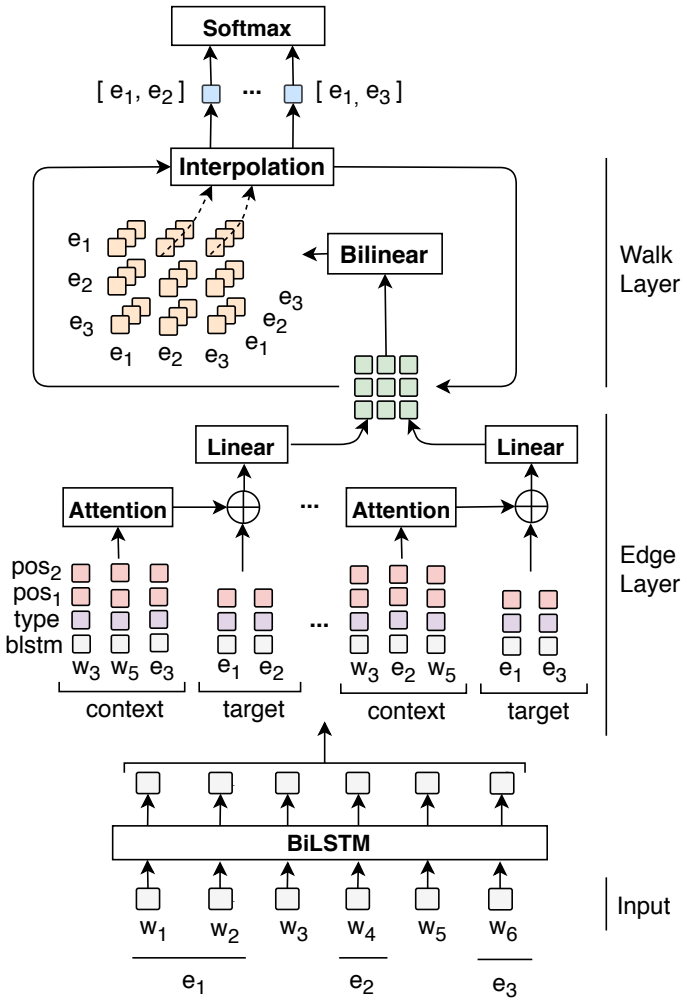


Figure 4.2: Overview of the proposed model. Each small square represents a vector. The relative position embeddings (pos_1, pos_2) and the semantic entity types (type) are generated in the embedding layer, but attached to each word or entity after the BiLSTM encoder. This is not explicitly shown in the figure for readability. The BiLSTM encoder receives only word embeddings.

4.2.1 Sequence Encoding

Sequence encoding is responsible for generating contextualised word representations, based on the input sentence.

Generation of embeddings

In order to encode the input sequence using neural models, we first need to map existing information into real-values, fixed dimensional vectors. This procedure takes place in an embedding layer, as described in Chapter 3.

For the proposed model, the embedding layer involves the creation of d_w, d_t, d_p-dimensional vectors which are assigned to words, semantic entity types and relative positions to the target entities, respectively. We map all words and semantic types into real-valued vectors $\mathbf{w}$ and $\mathbf{t}$. Relative positions to target entities are created based on the position of words in the sentence, motivated by the work of Zeng et al. (2014) and their effectiveness in RE. In more detail, for each word in the sentence, we attach two relative positions with respect to each target entity. For instance, as shown in Figure 4.3, the relative position of *surrendered* to *Basra* is $+4$ and the relative position to *troops* is -3. Similarly to words and semantic entity types, we embed real-valued vectors $\mathbf{p}$ to these positions. In case the target entities consist of more than one word,

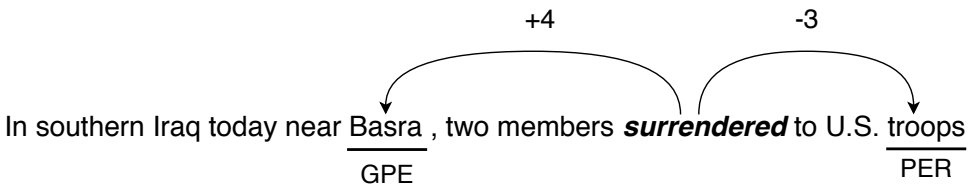


Figure 4.3: Example showing the relative positions of the word *surrendered* with respect to the target entities *Basra* and *troops*.

the relative positions are estimated using their first or the last word, depending on whether the context word (or entity) is before or after the target entity.

In essence, positional information further enables the model to learn explicitly where each word is positioned in the sentence. Semantic entity types provide a strong inductive bias for several classes, as has been shown in previous work (Zhang et al., 2006b). This additional information is not necessary for the model to work and we experiment in Section 4.5.3 with their contribution (or not) to the performance.

Contextualisation of words

In order to contextualise the word embeddings so that each word contains information about the words that surround it, we feed the previously defined word representations of each sentence into a BiLSTM layer, which was introduced in Chapter 3.3.2. BiLSTMs have been widely used for relation extraction (Miwa and Bansal, 2016; Zhou et al., 2016b; Katiyar and Cardie, 2017; Bekoulis et al., 2018b; Sahu and Anand, 2018) due to their effectiveness in encoding long-term dependencies between words. They additionally consider the sequential structure of the sentence, taking the word order into account. It is important to note, at this point, that the focus of the proposed approach is not on the choice of the encoder but, rather, on how to model the interactions between different pairs in a sentence. As such, one can replace this encoder with another one, e.g. Convolutional Neural Networks (Chapter 3.3.1) or Transformers (Vaswani et al., 2017).

For each word w_i in a sentence, we transform it into a word embedding $\mathbf{w}_i$ and feed it to the BiLSTM. We concatenate the two resulting representations from the left-to-right and right-to-left passes of the network into a d_e-dimensional vector, $\mathbf{x}_i = [\overrightarrow{\mathbf{h}_i}; \overleftarrow{\mathbf{h}_i}]$, which serves as the BiLSTM layer output of word w_i.

As already discussed, existing models developed for Relation Extraction assume that only one entity pair exists in a sentence. In previous approaches (Nguyen and Grishman, 2015; Miwa and Bansal, 2016; Sorokin and Gurevych, 2017), the input sequence is formed as the concatenation of word $\mathbf{w}$ embeddings and other information, such as semantic entity type $\mathbf{t}$ and/or relative position $\mathbf{p}$ embeddings. However, as we choose to treat all named entity pairs simultaneously, we cannot consider the concatenation of these embeddings as the encoder's input (particularly position). One named entity might participate in multiple relations, hence having different relative positions to the words in the sentence. We thus avoid encoding target pair-dependent information in the BiLSTM layer, and instead use only word embeddings as the encoder's input. This decision has two main advantages: (i) The computational cost is reduced, as this computation is repeated based on the number of sentences, instead of the number of pairs (i.e. each sentence is given to the encoder once, in comparison with other models that repeat each sentence based on the number of pairs (Nguyen and Grishman, 2015; Miwa and Bansal, 2016; Sorokin and Gurevych, 2017)), (ii) we share the sequence encoding layer's parameters among the pairs of a sentence. The second advantage is particularly important, as it enables the model to indirectly learn latent dependencies

between entities in the same sentence, during training. This is something that is confirmed by the work of Bekoulis et al. (2018b), in a multi-task setting when detecting both entities and relations. As we will show in the Section 4.5.2, sharing the sequence encoding layer across pairs in a sentence, resulted in large improvement when detecting relations between single pairs in a sentence.

4.2.2 Edge Layer

After constructing contextualised word representations for each sentence, the next step involves the construction of the graph. We first map the named entities into nodes and then consider their interactions as edges. For this particular setting, as we cannot know the related named entities in advance, nor their direction, we create a fully connected graph structure without self-node connections. Our graph representation is edge-oriented in the sense that it creates an edge representation for each pair, which includes information about the pair's nodes and context. This decision is based on the observation that since one entity can participate in multiple relations (with other entities), the connections of different pairs should be unique and not shared, as is the case of node-oriented models (Kipf and Welling, 2017). It is, thus, more straightforward to have different relation (edge) representations for each pair in a sentence, hence we investigate such an approach. This is also more suitable for the creation of walks, as will be later discussed.

In order to create pair-centric representations, which are associated with edge representations in our graph structure, we construct two types of information: (i) target pair representations and (ii) target pair context representations.

Target pair representations. If an entity e_j consists of I words, we create a single word representation by averaging the BiLSTM representations of its corresponding words,

$$\mathbf{x}_{e_j} = \frac{1}{|I|} \sum_{i \in I} \mathbf{x}_i, \tag{4.1}$$

where I is a set with the word indices inside entity e.

The final representation of an entity e_j is the concatenation of its surface form representation $\mathbf{x}_{e_j}$, the representation of its entity type $\mathbf{t}_{e_j}$ and the representation of its relative position to entity e_i: $\mathbf{p}_{e_j,e_i}$. Hence, the representations of a pair's entities (vertices) $\mathbf{v}_{e_i}$

and $\mathbf{v}_{e_j}$ are formed as follows:

$$\mathbf{v}_{e_i} = [\mathbf{x}_{e_i}; \mathbf{t}_{e_i}; \mathbf{p}_{e_i,e_j}], \tag{4.2}$$

$$\mathbf{v}_{e_j} = [\mathbf{x}_{e_j}; \mathbf{t}_{e_j}; \mathbf{p}_{e_j,e_i}], \tag{4.3}$$

where ';' corresponds to the vertical concatenation operation. In this equation, as well as in all of the following equations, we assume column vectors.

Target pair context representations. The next step involves the construction of the representation of the context for each target pair. The context of a target pair can be defined as all of the words and named entities in the sentence that are not part of the pair's arguments. Although the target pair representation already contains contextual information obtained from the BiLSTM layer, we choose to explicitly model the context of a particular pair in an effort to highlight words that are particularly important for the relation of the pair.

The context of a target pair consists of words (not belonging to any entity) and other entities in the sentence. For each context word w_z of a target pair e_i, e_j, we concatenate its BiLSTM representation $\mathbf{x}_{w_z}$, its semantic type representation $\mathbf{t}_{w_z}$ and two relative position representations: to target entity e_i, $\mathbf{p}_{w_z,e_i}$ and to target entity e_j, $\mathbf{p}_{w_z,e_j}$. These context words, that are not part of a named entity (w_z), are assigned a special semantic entity type "O" (out-of-entity). The final representation for a context word w_z of a target pair e_i, e_j is formed as,

$$\mathbf{v}_{e_i,e_j,w_z} = [\mathbf{x}_{w_z}; \mathbf{t}_{w_z}; \mathbf{p}_{w_z,e_i}; \mathbf{p}_{w_z,e_j}] \tag{4.4}$$

In a similar manner, the representation of a context named entity e_z (that is not one of the target entities) is formed as,

$$\mathbf{v}_{e_i,e_j,e_z} = [\mathbf{x}_{e_z}; \mathbf{t}_{e_z}; \mathbf{p}_{e_z,e_i}; \mathbf{p}_{e_z,e_j}] \tag{4.5}$$

It is important to note here that additional named entities in the sentence are treated as entities in the pair context.

For a sentence s, the context representations for all entity pairs are expressed as a fourth order tensor $\mathbf{C}$, where rows and columns correspond to entities and the depth corresponds to the context (words and entities) of the pair (n_c). These representations are then compiled into a single context representation for each pair, using an attention

mechanism.

One attention mechanism for RE was proposed in Zhou et al. (2016b), where attentive pooling was used to create a weighted average of the word representations in a sentence. We adapt the same mechanism on the context of each pair as follows,

$$\begin{aligned} \mathbf{u} &= \mathbf{q}^\top \tanh(\mathbf{C}_{e_i,e_j}), \\ \alpha &= \text{softmax}(\mathbf{u}), \\ \mathbf{c}_{e_i,e_j} &= \mathbf{C}_{e_i,e_j}\, \alpha^\top, \end{aligned} \tag{4.6}$$

where $d_c = d_e + d_t + 2d_p$ is the dimensionality of the context representations, $\mathbf{C}_{e_i,e_j} \in \mathbb{R}^{d_c \times n_c}$, $\mathbf{q} \in \mathbb{R}^{d_c}$ denotes a trainable attention vector, $\alpha \in R^{1 \times n_c}$ is the attention weights vector and $\mathbf{c}_{e_i,e_j} \in \mathbb{R}^{d_c}$ is the context representation of the pair e_i, e_j. An illustration of the computation of the above procedure is depicted in Figure 4.4.

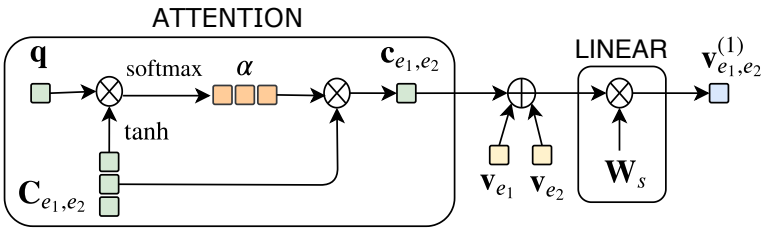


Figure 4.4: Attention and Linear layers used in the network.

Finally, we concatenate the representations of the target pair and the pair's context. We use a fully connected linear layer to reduce the dimensionality of the resulting vector, as follows,

$$\mathbf{v}^{(1)}_{e_i,e_j} = \mathbf{W}_s\, [\mathbf{v}_{e_i}; \mathbf{v}_{e_j}; \mathbf{c}_{e_i,e_j}] \in \mathbb{R}^{d_s}, \tag{4.7}$$

where, $d_m = 2d_e + d_c$ is the dimensionality before reduction, $\mathbf{W}_s \in \mathbb{R}^{d_s \times d_m}$ with $d_s < d_m$ and d_s is the dimensionality after reduction. This vector corresponds to the representation of the initial edge between vertices i and j. The (1) in the exponent indicates that the edge contains information for this pair only.

4.2.3 Walk-based Inference

Our principal aim is to support the relation between an entity pair using interactions with co-existing entities in the same sentence. We model these relations as chains of associations from the first to the second target entity, via other entities. To model such

chains of interactions, we propose a walk-based inference mechanism that iteratively generates and aggregates multi-hop representations.

We initially define the concepts of paths and walks from Graph Theory. Let $G = (V, E)$ be a graph with V vertices and E edges. In Graph Theory, a *path* is a set of edges that connect a sequence of distinct nodes. On the contrary, a *walk* is a set of edges that connect a sequence of nodes, though nodes can be repeated. The reasons that we choose walks instead of paths are two-fold. Firstly, walks are more flexible as there is no need to keep track of repeated nodes. Secondly, walks have been used by some methods that operate on graphs for construction of node embeddings (Perozzi et al., 2014; Grover and Leskovec, 2016) with successful results. However, we should highlight that these approaches incorporate random walks, while our search is not completely random, as will be explained later.

With this layer we generate a representation for each target entity pair. This representation encapsulates a finite number of walks (interaction chains) of different lengths between the arguments of the pair. The representation of one-length walk between two entities (Equation (5.8)) serves as a building block, in order to create representations for one-to-l length walks between the pair arguments in an iterative manner. For notation simplicity in the following equations, we consider entity vertices i and j instead of e_i and e_j. The walk-based algorithm can be seen as a two-step process repeated N times: *walk generation* and *walk aggregation*.

Walk generation. During the generation step, our goal is to combine two edges in order to form a new one. Our focus is to create edge representations, instead of node representations, since we aim to represent the latent association between two entities in the form of a unique vector. We take inspiration from the work of Socher et al. (2013), who proposed a neural tensor network to identify whether a certain relation between two named entities holds. Their approach is compositional, i.e. two representations are combined, forming a new one. In their approach, they use a bilinear transformation to combine two entity representations with a relation-specific matrix, one for each relation, into a vector. We follow a similar procedure applied on the edges, i.e. we combine two edge representations into a new one. We choose to simplify the relation matrix into a second-order tensor, in order to enable the sharing of commonalities between relation types. Moreover, we modify the computation to one element-wise multiplication and one tensor multiplication, in order for the composition function to result into a vector of same dimensionality as the inputs. This is particularly useful

when iterating over the walk layer several times. The new vector is then passed from a non-linear activation function, as follows,

$$f\left(\mathbf{v}_{ik}^{(l)}, \mathbf{v}_{kj}^{(l)}\right) = \sigma\left(\mathbf{v}_{ik}^{(l)} \odot \left(\mathbf{W}_b\, \mathbf{v}_{kj}^{(l)}\right)\right), \tag{4.8}$$

where $\mathbf{v}_{ik}^{(l)} \in \mathbb{R}^{d_s}$ corresponds to an edge representation that includes information from walks of length one-to-l between entities e_i and e_k, $\odot$ represents element-wise multiplication, σ is the sigmoid non-linear function and $\mathbf{W}_b \in \mathbb{R}^{d_s \times d_s}$ is a trainable weight matrix. This equation results in edge representations that include information from walks of length two-to-$2l$. An edge representation created as such is, essentially, the representation of a walk from the first entity node i to the second entity node j via some intermediate node k.

Walk aggregation. In the walk aggregation step, we aim to combine shorter and longer walks from the first target node to the second, in order to include all possible walk lengths inside the pair representation. We, thus, linearly combine the initial edge representations and the newly composed edge representation using linear interpolation,

$$\mathbf{v}_{ij}^{(2l)} = \beta\, \mathbf{v}_{ij}^{(l)} + (1-\beta) \sum_{k \neq i,j} f\left(\mathbf{v}_{ik}^{(l)}, \mathbf{v}_{kj}^{(l)}\right), \tag{4.9}$$

where β is a scalar that indicates the importance of the representations that include shorter walks.

We generally consider β as a hyper-parameter, which we tune. Overall, we expect that performance will increase with a larger β for shorter walks based on the shortest path assumption. In detail, the shorted path has been used on dependency trees (Xu et al., 2015c), but also node classification (Borgwardt and Kriegel, 2005), where it was proved that longer walks reduce performance. We hypothesise that this can be extended for walks generated as chains of interactions between entities.

Iteration. Overall, using Equations (4.8) and (4.9) with $l = 1$, we create edge representations encoding walks of length one-to-two. We then create walks of length one-to-four by re-applying these two equations in order with $l = 2$. We repeat this process a finite number of times, N. The maximum length of the generated walks is equivalent to $L = 2^N$. Finally, if a sentence contains only two named entities, we do not use the walk-based mechanism. Instead, we force the pair to keep its original representation.

In brief, the iterative algorithm of the walk-based inference is described below:

Algorithm 1 Walk-based inference iterative algorithm.

Require: V vertices, E edges, N iterations
Ensure: $l = 1, n = 1$
for $n \leq N$ **do**
  for $i, j \in V \times V$ **do**
    for all $k \in V$ and $k \neq i, j$ **do**
      combine $v_{ik}^{(l)}$ and $v_{kj}^{(l)} \rightarrow v_{ikj}^{(2l)}$
    end for
    aggregate via $k \rightarrow v_{ij}^{(2l)}$
  end for
  $l = 2^n$
  $n = n + 1$
end for
return $v_{ij}^{(2^N)}$

4.2.4 Classification

In the final layer of the network, we pass the resulting pair representation into a fully connected linear layer with a softmax function on top to generate normalised probability scores for each relation category.

$$\mathbf{y} = \text{softmax}(\mathbf{W}_r \, \mathbf{v}_{e_i,e_j}^{(L)} + \mathbf{b}_r), \tag{4.10}$$

where $\mathbf{W}_r \in \mathbb{R}^{d_r \times d_s}$ is the weight matrix, d_r is the total number of relation types and $\mathbf{b}_r \in \mathbb{R}^{d_r}$ is the bias vector.

As mentioned earlier, we do not know the direction of the relation between two named entities, i.e. whether the relation is from the first argument to the second or the opposite. For this reason, we augment the number of relation categories by the inverse relations. In essence, if a relation is formed from the first argument to the second, we name it 1-to-2, and when the relation is formed from the second argument to the first, we name it 2-to-1. In total, we use $2d_r + 1$ relation categories. The additional category corresponds to the *no relation* (NR) category.

4.3 Experimental Settings

4.3.1 Datasets and Comparisons

In order to test the performance of the proposed model, we choose to evaluate it on four, sentence-level datasets specific on the generic domain.

ACE 2004 defines 7 coarse-grained semantic entity categories (*Facility* (FAC), *Geo-Political Entities* (GPE), *Location* (LOC), *Organisation* (ORG), *Person* (PER), *Vehicle* (VEH) and *Weapon* (WEA)) and 7 relation categories (*Artifact* (ART), *EMP-ORG*, *GPE-AFF*, *Other-AFF*, *Person-Social* (PER-SOC), *Physical* (PHYS)) (Doddington et al., 2004). We follow the same process as Miwa and Bansal (2016) by removing the *disc* domain documents, using the same pre-trained word embeddings and doing2 5-fold cross-validation on *bnews* and *nwire* domains. The statistics of the dataset are shown in Table 4.1. For this dataset, we aim to not only identify the relation type between an entity pair, but also the directionality of the relation.

We compare our model with *SPTree* Miwa and Bansal (2016) using the same pre-trained Wikipedia word embeddings and data split. In order to fairly compare our model with theirs, we re-trained the latter using the publicly available source code on gold named entities, with early stopping with patience equal to five.

	Data
Documents	
Sentences	4,951
Entities	21,509
Positive Pairs	4,084
ART	211
EMP-ORG	1,624
GPE-AFF	529
OTHER-AFF	142
PER-SOC	365
PHYS	1,213
Negative pairs (%)	91.90
Average sentence length	24
Average entities/sentence	4.3

Table 4.1: Statistics for ACE 2004 dataset.

	ACE05-D			ACE05-ND
	Train	Dev.	Test	
Documents	351	80	80	595
Sentences	5,417	1,342	1,128	10,922
Entities	24,410	5,909	5,087	48,153
Positive Pairs	4,780	1,131	1,151	8,669
ART	489	96	151	872
GEN-AFF	511	124	104	950
ORG-AFF	1,469	365	359	2,516
PART-WHOLE	774	162	182	1,299
PER-SOC	438	106	77	1,085
PHYS	1,099	278	278	1,947
Negative pairs (%)	92.12	92.16	90.98	92.72
Avg. sentence length	22	21	22	22
Avg. entities/sentence	4.5	4.4	4.5	4.4

Table 4.2: Statistics for ACE 2005 dataset for two different settings, *ACE05-D*: classification of relation type and direction and *ACE05-ND*: classification of relation type only.

ACE 2005 is an improved version of ACE 2004. It defines the same 7 semantic entity categories and 6 relation categories (*Artifact* (ART), *Gen-Affiliation* (GEN-AFF), *Org-Affiliation* (ORG-AFF), *Part-Whole* (PART-WHOLE), *Person Social* (PER-SOC) and *Physical* (PHYS)).

We compare our model with two state-of-the-art models on this dataset. The first model, is again *SPTree* (Miwa and Bansal, 2016). We follow the same pre-processing,

removing the *cts*, *un* domain documents, use the same train/development/test data split and the same pre-trained word embeddings. The statistics of the dataset are shown in Table 4.2. Again, we re-trained the model with gold named entities and applied early stopping with patience equal to five. For this version of ACE 2005, we aim to identify both the relation category and the directionality of the relation between two named entities. We will refer to this version as ***ACE05-D*** (direction).

The second model, is named *CNN* (Nguyen and Grishman, 2015). We follow the same setting, performing 5-fold cross validation (the folds were provided by the authors) on the dataset and used the same pre-trained word embeddings. We additionally remove entity type embeddings since they were not used in their work. The authors removed pairs with distance larger than 15 words, hence reducing the number of negative pairs in the dataset by 2%. Instead, we keep these pairs during both training and prediction. The dataset statistics are shown in the last column of Table 4.2.
For this version of ACE 2005, we do not identify the direction of the relation; we merely predict the semantic relation type between two named entities, following Nguyen and Grishman (2015). We will refer to this version as ***ACE05-ND*** (no direction).

WikiData is a distantly supervised dataset developed by Sorokin and Gurevych (2017) using the Wikipedia corpus. The dataset contains sentences with multiple named entities and consists of 351 relation categories. It is split into training, validation and held-out sets without overlapping in terms of sentences and relation instances. We slightly modified the dataset as follows: We removed self-relations, duplicated pairs and pairs that have at least one missing argument. These changes led us to use approximately 99.7% of the original dataset, as shown in Table 4.3.

To draw comparisons on this work, we compare with the *ContextAware* model of Sorokin and Gurevych (2017), using the same data split, and pre-train Glove word embeddings. We also remove semantic entity type embeddings for a fair comparison, since they were not used in their work. We re-ran the ContextAware model on the modified dataset and report on the performance. Two minor differences between the two models are: firstly, the authors treat only up to 7 pairs in a sentence at the same time and, secondly, they restrict the maximum sentence length to 36. Our proposed model does not contain these restrictions. We do not classify different directionality candidates for this dataset. Instead, we consider the direction of each pair as given during classification, similar to their work.

	Training	Validation	Held-out
Original # pairs	777,481	252,277	740,963
Missing arguments	985	273	962
# Self-pairs	577	202	593
# Duplicate pairs	5	1	12
Final # pairs	775,914 (99.7%)	251,801 (99.8%)	739,396 (99.7%)
# Sentences	371,860	123,751	360,083
# Positive pairs	547,763	178,291	573,013
# Negative pairs	228,151	73,510	166,383
Average sentence length	23.74	23.72	23.91
Average entities/sentence	3.14	3.08	2.94

Table 4.3: Statistics for the WikiData dataset.

SemEval 2010-Task 8 is a manually annotated, sentence-level dataset with 9 relation categories (*Cause-Effect*, *Instrument-Agency*, *Product-Producer*, *Content-Container*, *Entity-Origin*, *Entity-Destination*, *Component-Whole*, *Member-Collection* and *Message-Topic*) (Hendrickx et al., 2010). Each sentence contains only two named entities. We consider the *Other* relation category as the negative relation category. The dataset is split in $8,000$ training and $2,717$ test sentences. We use 800 randomly selected sentences from the training set to act as our development set. The statistics of the dataset are shown in Table 4.4. The official performance metric is macro-averaged F1-score. We will refer to the dataset as ***SE-2010*** for brevity.
Although this dataset does not have multiple entities per sentence, we choose to evaluate our model on it in order to show that it can still work in such datasets, even without using the walk-based mechanism. We directly compare our model with Miwa and Bansal (2016) using the same pre-processing and pre-trained Wikipedia word embeddings.

4.3.2 Implementation Details

For all datasets, the Stanford CoreNLP toolkit (Manning et al., 2014) was used for sentence splitting and tokenisation. Pairs are generated between all entities residing in a sentence. Pairs that are not assigned a semantic relation category in the annotations receive the NR (no-relation) category.

We originally implemented our model using the Chainer library[a] (Tokui et al., 2015). In this dissertation, we report the results which can be obtained by using the

[a] https://chainer.org/

	Train	Dev.	Test
Sentences	7,200	800	2,717
Entities	14,400	1,600	5,434
Positive Pairs	5,933	657	2,263
Cause-Effect	901	102	328
Component-Whole	859	82	312
Content-Container	486	54	192
Entity-Destination	761	84	292
Entity-Origin	641	75	258
Instrument-Agency	448	56	156
Member-Collection	631	59	233
Message-Topic	573	61	261
Product-Producer	633	84	231
Negative pairs (%)	17.6	17.88	16.71
Average sentence length	19.09	18.98	19.12

Table 4.4: Statistics for SemEval-2010 Task 8 dataset.

published version of the model written in PyTorch[b]. The model was trained with the Adam optimiser (Kingma and Ba, 2015) using cross-entropy loss (Golik et al., 2013). In the following experiments we assume entities are given and use existing pre-trained word embeddings.

Regarding more specific network settings, the forget bias of the BiLSTM layer was initialised with a value equal to one, following the work of Jozefowicz et al. (2015). Gradient clipping, dropout on the embedding and output layers and L2 regularisation, without regularising the biases, are incorporated to avoid overfitting. We additionally use early stopping in order to chose the number of training epochs and use parameter averaging following Hashimoto et al. (2013) during prediction. Similarly to previous work, we observed that averaging the parameters resulted in more stable training. We tune the model hyper-parameters on the development set of the ACE05-D version, using the RoBO Toolkit (Klein et al., 2017). We use the same hyper-parameters on the ACE05-ND version and ACE 2004. The only hyper-parameter that is tuned separately for this dataset is the best training epoch, based on the development set. For the WikiData and SemEval datasets, we tune hyper-parameters on the respective development sets. Readers can refer to Appendix A.1 for the detailed hyper-parameter settings of the proposed model on each dataset.

[b] https://github.com/fenchri/walk-based-re

4.4 Results

Evaluation is performed on all datasets by reporting the performance based on the primary metric (micro- or macro-average) Precision, Recall and F1-score of each dataset. In more detail, we report two models. A *baseline* model ($L = 1$) that does not consider the walk-based layer; that is, the pair representation is simply formed as the concatenation of the pair entities and their context. The second model incorporates the walk-based layer. We experiment with a different number of maximum walk lengths L (2, 4, 8). The model with the best performing walk length on the development is used as the primary (chosen) model on the test set. However, we also report the performance on the test set with other walk lengths, in order to better illustrate their effect.

For all the following experiments, we perform significance testing using the Approximate Randomisation algorithm (Noreen, 1989).

4.4.1 Candidate Pairs Classification

Before moving on to the main results, we report some preliminary experiments regarding the choice of pair candidates for classification. As we mentioned in Section 4.2.4, the number of relation categories is augmented with the inverse relation types in order to represent directionality of relations. In addition, our entity graph is fully connected and non-directed since we do consider any restrictions regarding the connections of entities in a sentence. As a result, the proposed model generates two edge representations for each pair, i.e. one from the first argument to the second and the inverse. These representations are substantially different from one another since in the walk generation process we introduced a non-linear activation function (Equation (4.8)). It is thus reasonable to experiment with which of these two representations is better to choose for classification.

In our initial experimentation (Christopoulou et al., 2018), we followed the setting of Miwa and Bansal (2016), where candidate pair instances were generated based on the position of their arguments in the sentence. In a left-to-right instance (L2R) the first argument appears first in a sentence and the second appears second, while the opposite applies in a right-to-left instance (R2L). For example, in the sentence of Figure 4.1, for the target pair *Basra-troops*, a L2R instance is (*Basra*, *troops*) with relation 2-PHYS-1 and a R2L instance is (*troops*, *Basra*) with relation 1-PHYS-2. Following previous work, we classified both edge representations (corresponding to L2R and

R2L instances) and summed their respective losses (Loss = L2R+R2L). The final relation label for a target pair was selected by choosing the positive (i.e. different from NR category) and most confident prediction between the two, as in Xu et al. (2015b) (Prediction = L2R+R2L). The performance of this experiment, on the ACE05-D development set, can be seen in the first row of Table 4.5 (setting A).

			Micro F1 (%)			
Setting	*Loss*	*Prediction*	$L = 1$	$L = 2$	$L = 4$	$L = 8$
A	L2R+R2L	L2R+R2L	59.51	62.09	62.77	60.87
B	**L2R**	**L2R**	61.50	63.09	64.80*	63.13*
C	**R2L**	**R2L**	61.94*	63.19	65.10*	64.09*
D	L2R+R2L	L2R	58.44	61.65	62.41	60.69
E	L2R+R2L	R2L	59.62	61.36	62.38	60.65
F	L2R+R2L	L2R+R2L†	59.64	61.88	62.58	61.05
G	L2R	L2R+R2L	54.10*	61.28	63.40	61.79
H	R2L	L2R+R2L	51.73	62.15	63.67	60.49
I	Ensemble E & F		61.81	63.03	65.10*	63.72*

Table 4.5: Performance of different pair candidates on the ACE05-D development set. $*$ indicates significance at $p < 0.05$ in comparison with setting A. † indicates that we choose the most confident prediction after classifying both instances.

We later experimented with training the model using only information from one of the two instances (settings B, C), where we observed significant performance improvement in comparison with our original setting A. This is contradictory with the findings of Miwa and Bansal (2016), where the authors observed that using one of the instances produced similar performance to using both.

In order to analyse the reason for this phenomenon, we first evaluated different prediction settings to make sure that the low performance of setting A is not because of errors when resolving conflicts between predictions. Settings D and E correspond to choosing one of the two instances for prediction instead of resolving them. In setting F we always choose the prediction of the most confident instance. As it can be observed from Table 4.5, all of these settings produced similar performance to setting A, indicating the this phenomenon is potentially related with how we compute the loss (joint training of both instances). Indeed, when computing the loss over one instance but using two predictions (settings G, H) performance is significantly worse for the $L = 1$ model, but this is not the case for the walk-based model. We speculate that this is an indication that the walk-based layer captures directionality information, since it can classify both instances well.

We attribute this behaviour, in comparison with the finding of Miwa and Bansal (2016), to the following (plausible) reasons. In the model of Miwa et al. (2009), the SDP between two entities with dependency edge labels is used to represent the pair. The SDP contains strong biases regarding the correct direction of the relation. On the contrary, our model (before training) has no prior knowledge about the potentially correct relation direction. The propagation of information from instances of different directionalities can confuse the model that does not have any guiding relation direction biases. Possibly, adding further constrains to the classifier can alleviate this. However we leave further investigation as future work. We conclude that, for our proposed model, joint training of instances with different directions hurts the performance and instead one should be used. For the remaining datasets, in case we classify the direction of the relation, we experiment with both instances and choose those one that yielded the best results on the development set. By applying the AR significance test, the difference between the two is not significant (settings B, C).

4.4.2 Performance Comparison

Table 4.6 illustrates the performance of our proposed model in comparison with state-of-the-art models on four different datasets.

In all datasets, the walk-based model with 4-length walks ($L = 4$) is significantly better than the baseline $L = 1$. Regarding the ACE05-D dataset, we compare our model with SPTree (Miwa and Bansal, 2016), where higher performance was achieved, though not to a significant degree. We additionally compare with a CNN model (Nguyen and Grishman, 2015), where again the best performing model was $L = 4$. Regarding the performance on the ACE 2004 dataset, our proposed model outperforms SPTree model. We also manage to outperform the model of Sorokin and Gurevych (2017) on the distantly supervised WikiData dataset by a significant margin for all walk lengths, with best results for $L = 4$. For the SemEval 2010 dataset, our model cannot outperform the state-of-the-art but we can achieve a decent performance when using the baseline model.

The difference in performance between the ACE datasets and the rest (WikiData, SE-2010) is due to the number of negative samples. The ACE datasets contain approximately 92% of no relation pairs, while WikiData and SE-2010 have approximately 30% and 17% respectively. As a result, finding related pairs in the ACE datasets is much more challenging compared to the other two. In addition, in the WikiData dataset, directionality is assumed given, which improves performance by almost 10%

Dataset	Model	Dir.	Metric	P (%)	R (%)	F1 (%)
ACE05-D	SPTree (Miwa and Bansal, 2016)	✓	Micro	69.4	62.0	65.5
	No walks $L = 1$			72.7	57.9	64.5
	+ walks $L = 2$			75.3	58.1	65.6
	+ walks $L = 4$			72.1	62.2	**66.8***
	+ walks $L = 8$			71.8	59.5	65.0
ACE05-ND	CNN (Nguyen and Grishman, 2015)	×	Micro	71.5	53.9	61.3
	No walks $L = 1$			69.8	52.8	60.1
	+ walks $L = 2$			74.0	53.3	61.9
	+ walks $L = 4$			69.8	56.1	**62.2***
	+ walks $L = 8$			69.5	53.1	60.2
ACE 2004	SPTree (Miwa and Bansal, 2016)	✓	Micro	64.3	60.5	62.3
	No walks $L = 1$			69.3	61.1	64.9$^{\diamond}$
	+ walks $L = 2$			71.0	61.4	65.8$^{\diamond}$
	+ walks $L = 4$			70.3	62.4	**66.1**$^{*\diamond}$
	+ walks $L = 8$			72.3	60.0	65.5$^{\diamond}$
WikiData	ContextAware (Sorokin and Gurevych, 2017)	×	Micro	79.9	77.8	78.9
	No walks $L = 1$			82.0	74.5	78.1$^{\diamond}$
	+ walks $L = 2$			81.9	76.5	79.1$^{*\diamond}$
	+ walks $L = 4$			81.9	77.8	**79.8**$^{*\diamond}$
	+ walks $L = 8$			81.5	77.6	79.5$^{*\diamond}$
SE-2010	SPTree (Miwa and Bansal, 2016)	✓	Macro	82.2	87.4	84.7
	CNN (Nguyen and Grishman, 2015)			-	-	82.8
	No walks $L = 1$			79.4	83.5	81.4

Table 4.6: Performance on the four datasets in comparison with the state-of-the-art. $*$ indicates significance at $p < 0.05$ in comparison with $L = 1$. $\diamond$ indicates significance at $p < 0.05$ in comparison with the state-of-the-art. *Dir* indicates identification of relation direction or not.

in preliminary experiments. This is not reflected in the ACE05-ND results, as this performance boost is counterbalanced by the removal of semantic entity type embeddings. Type embedding contribute approximately 8% to the performance, as will be discussed in Section 4.5.3.

4.5 Analysis and Discussion

We choose ACE05-D as the primary dataset to conduct extensive analysis; in particular, we employ the data split of Miwa and Bansal (2016). The reasons for this choice are attributed to the manual annotation of the dataset, the significant improvement in terms of relation categories and general annotation quality (Li and Ji, 2014) over ACE 2004 and that identifying both relations and their directions is a more challenging task.

Additionally, a data split as such is preferred due to computational restrictions compared with a cross-validation setting. We also report some basic analysis on the WikiData dataset as a large multi-entity dataset. However, due to its automatic generation, we do not consider deeper analysis.

We divide our analysis into three parts. The first part discusses particular errors of the model, using both quantitative and qualitative analysis. The second pertains to the walk-based layer, i.e. analysis that can evaluate the effectiveness of this mechanism and in which cases it performs better than other models. The last part discusses additional model enhancements, such as the effect of positional and semantic entity type embeddings as well as the efficacy of the attention mechanism in the construction of the context representation.

4.5.1 Error Analysis

We investigate the performance of our model on different relation categories in comparison with the state-of-the-art. As it can be observed from Table 4.7, the walk-based model outperforms SPTree only in the *ORG-AFF* and *ART* relation categories of the ACE05-D dataset. Two plausible reasons exist for this behaviour. First, the SPTree

Category	F1 (%)				
	$L=1$	$L=2$	$L=4$	$L=8$	SPTree
PHYS	46.32	43.84	51.90	48.95	52.95
ORG-AFF	79.71	81.13	81.87	79.32	78.22
GEN-AFF	58.48	56.25	54.65	54.86	56.00
ART	57.94	58.33	60.63	56.56	49.36
PART-WHOLE	64.48	68.89	68.85	68.89	71.66
PER-SOC	70.34	70.59	65.75	66.67	69.80
Micro score	64.51	65.62	66.85	65.08	65.53
Macro score	63.19	64.10	64.37	63.03	63.75

Table 4.7: Performance for each class on the ACE05-D test set for multiple walk lengths. *SPTree* refers to the model proposed by Miwa and Bansal (2016).

model uses a dependency parser, which enables the detection of patterns that indicate particular relationships with high accuracy. On the contrary, in our model we rely on the attention mechanism in order to identify informative words for the pair. While this has the advantage that words ignored by the SDP can now be considered, particular relations exhibit certain syntactical structures that are more easily captured by dependency-based models.

As we can observe in some cases, the walk-based mechanism might not be required, thus resulting in lower performance. For instance, for the *PER-SOC* (Person-Social) and *GEN-AFF* relation categories, the baseline ($L = 1$) model performs better than, or comparably to, the walk-based models. As the pair representations are updated simultaneously for all the pairs in the same sentence, latent dependencies between entities in a sentence can also be encoded through the shared sequence layer, which explains the success of $L = 1$ in some cases. For example, in the *GEN-AFF* category, although walks are not useful, the baseline model performs better than SPTree.

We further investigate the confusion matrix of the best performing model $L = 4$ on the ACE05-D test set. As observed from Table 4.8, the majority of errors occur due to False Negatives (predicted class NR: *no relation*), which can be justified by the large portion of the negative relations in the dataset (approximately 92%).

true / pred	ART	GEN-AFF	ORG-AFF	PART-WHOLE	PER-SOC	PHYS	NR
ART	77	1	0	0	0	6	67
GEN-AFF	0	47	10	4	0	5	38
ORG-AFF	0	3	289	0	0	2	65
PART-WHOLE	0	2	0	126	0	4	50
PER-SOC	0	0	0	0	48	0	29
PHYS	0	4	0	4	0	130	140
NR	26	11	48	50	21	76	

Table 4.8: Confusion matrix on the ACE 2005 test set (*SPTree* split) for $L = 4$.

We perform similar analysis on the WikiData dataset, estimating the performance of the most frequent classes, similar to Sorokin and Gurevych (2017), as shown in Table 4.9. The walk-based model does not always outperform the ContextAware model. Our model is mostly helpful on the *Citizenship*, *Subclass of* and *Instance of* relation categories. Similar performance with the ContextAware model is observed for *Part of* and *Sport* relation categories. Overall, we can notice that the walk-based mechanism is beneficial in almost all relation categories in comparison to the Baseline ($L = 1$), indicating that, in sentences with multiple named entities, interactions between pairs are important for relation extraction. The difference between our model and the ContextAware model is that our approach aims to form chains of interactions between two named entities, through other entities, in the same sentence. The latter represents a pair as a weighted average of the representations of the context pairs residing on the same sentence. It thus mostly relies on modelling the co-occurrence of different relation types between pairs in the same sentence and, consequently, measures how similar two pairs are. We deem that these two approaches are complementary and potential

Category	F1 (%)				
	$L = 1$	$L = 2$	$L = 4$	$L = 8$	ContextAware
Located in	79.68	81.79	83.30	83.05	84.99
Shares border	69.04	70.40	71.99	72.04	75.35
Citizenship	91.96	92.09	92.21	92.19	90.54
Subclass of	56.49	58.82	59.04	57.41	48.42
Instance of	84.96	85.20	85.26	85.29	84.55
Part of	50.08	50.55	50.76	50.86	51.24
Country	82.65	84.72	86.59	86.32	90.37
Sport	98.18	98.08	98.15	98.22	98.19
Micro score	78.10	79.17	79.83	79.56	78.90
Macro score	76.86	77.96	78.60	78.35	78.04

Table 4.9: Performance for the top 8 most frequent relation categories on the WikiData test set. The Macro F1-scores correspond to these classes only. *ContextAware* refers to the model proposed by Sorokin and Gurevych (2017).

combination could lead to better performance.

We finally report some qualitative examples on ACE05-D and Wikidata datasets. In the first example, the model does not detect the relation between *Putin* and *Russia*.

Case	Pred.	SPTree	Truth	Sentence
Common sense	NR	NR	ORG-AFF	they should be very worried about this item , I believe **Putin** is slowly sliding **Russia** back to what it once was .
Semantics	PHYS	PHYS	NR	**He** is being tried in **Greece** in absentia .
Coordination	ART	ART	ART	it was hit by **coalition bombs** and *missiles* and then burned and looted by *iraqis* .
Coordination	ART	NR	ART	it was hit by **coalition** *bombs* and **missiles** and then burned and looted by *iraqis* .
Implicit relation	PHYS	NR	NR	*Tareq Ayyoub* , a **journalist** with *Al-Jazeera* , died when a *U.S* . *warplane* bombed the Arab-language satellite television 's **office** .

Table 4.10: Examples of predictions made by the best performing walk-based model $L = 4$ on the ACE05-D dataset. The named entities in **bold** indicate the target pair arguments. The named entities in *italics* indicate other entities in the sentence.

As observed, the ORG-AFF relation between the two entities is not implied by the sentence and the model needs to perform common-sense reasoning in order to correctly identify the relation. Currently, our model does not directly support this, and neither does SPTree, except from the usage of pre-trained word embeddings. However, even

if the original embeddings captured some correlation between these two entities, they are likely to lose their original meaning during model training. This can be because the original embeddings were trained on Wikipedia, but the ACE05-D dataset contains mostly news articles that refer to war, political disputes etc, that Wikipedia might not contain. Another difficult case is that of the second example. The entity *He* cannot be considered situated in *Greece*, due to the presence of the phrase *in absentia*, which indicates absence. However, both models incorrectly predict it as a *Physical* relation. This illustrates a larger problem of the models that do not take bigrams into account.

In the following two examples, the model correctly identifies the artifact relations, whereas SPTree is able to find only one of the two. The final example can be considered an implicit relation; that is, the relation is not directly expressed in the sentence, though we can infer it by reading it. The *journalist* cannot have died if he was not located in the *office*. Due to annotation restrictions by the ACE05 dataset, however, such relations are not annotated.

Case	Pred.	CA	Truth	Sentence
KB inc.	cast member	cast member	NR	She made her Hollywood feature film debut in **The Grudge 2** in 2006 , a horror sequel starring Amber Tamblyn and **Sarah Michelle Gellar** .
Distance	subclass of	NR	subclass of	It consists in the **worship** of the ngel zex , the Bai word for patrons or lords , rendered as benzhu in Chinese , that are local gods and **deified ancestors** of the Bai nation .
Coordination	prod. company	NR	prod. company	Famous American serials of the silent era include **The Perils of Pauline** and The *Exploits of Elaine* made by **Pathé Frères** and starring *Pearl White* .
Coordination	cast member	NR	cast member	Famous American serials of the silent era include *The Perils of Pauline* and **The Exploits of Elaine** made by *Pathé Frères* and starring **Pearl White** .

Table 4.11: Examples of predictions made by the best performing walk-based model $L = 4$ on the WikiData dataset. CA corresponds to the ContextAware model. The named entities in **bold** indicate the target pair arguments. The named entities in *italics* indicate other entities in the sentence.

Regarding the WikiData dataset, as shown in Table 4.11, a first category of errors for both our model and the ContextAware is that the KB that was used to create the corpus, is incomplete. As a result, even though several relations stand, they are not annotated in the dataset. Secondly, we observe from the second row of the table that our

model can detect relations that are very far apart (24 words), while the ContextAware cannot do so. Finally, we can once again see that our model performs well in coordination, particularly in cases were multiple entities exist and relate to each other. For instance, in the last two examples, ContextAware fails to find both relations, while the walk-based model not only assigns a positive relation, but also differentiates the relation between the arguments, despite not incorporating semantic entity types.

4.5.2 Walk-based mechanism

We then move on to analyse the walk-based mechanism. As a first attempt, we estimate the performance of the proposed model, for multiple walk lengths, on sentences that contain a varying number of entities. In more detail, we group sentences by considering the number of entities required to form complete hops with the walk-based model. For instance, when three entities exist in the sentence, we can form up to two-length paths $L = 2$. When four or five entities exist, we need $L = 4$ and for 6 to 9 entities we need $L = 8$. We expect that, for multi-entity sentences, the walk-based mechanism should perform better than the baseline model.

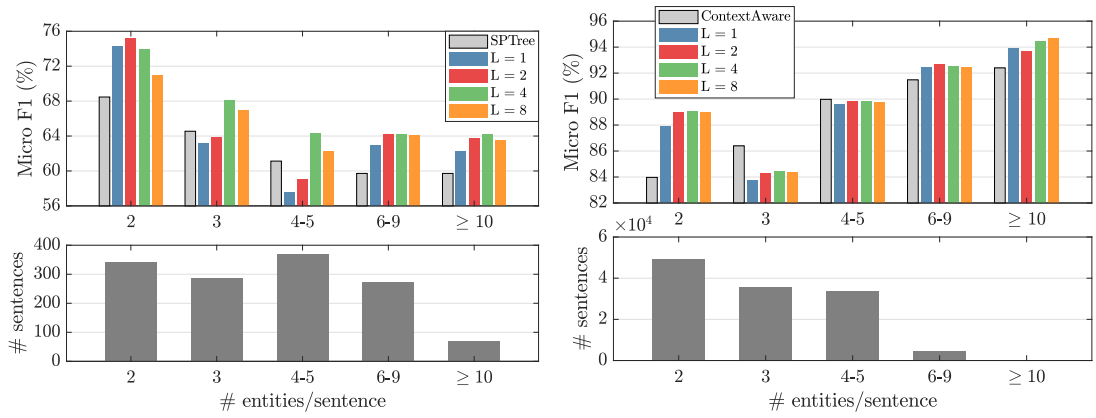


Figure 4.5: Performance as a function of the number of entities in a sentence for different number of walks on the (a) ACE05-D and the (b) WikiData development sets. The bar plots in the second row illustrate the distribution of sentences for each group of entities.

Results in Figure 4.5 reveal that, for multi-pair sentences, the walk-based model performs better than the baseline model ($L = 1$). However, the existence of multiple pairs does not indicate that all pairs can benefit from one another. For instance, for 6-9 entities, the performance of $L = 2, 4, 8$ is similar, indicating that more walks

are not necessarily beneficial despite the presence of more named entities in the sentence. This observation implies that, in general, the walk-based mechanism should be adapted to different pairs, using potentially different hops instead of using all possible intermediate entities in the sentence. For the WikiData dataset, we observe that the walk-based model is more effective, either for single pairs (similarly to the ACE05-D dataset) or for more than 6 entities. For three named entities, the ContextAware model outperforms our approach, implying that co-occurrence of relations is more common when fewer entities exist in the sentence. It is important to note that, during training, parameters are updated by considering all pairs simultaneously. As a result, latent dependencies between entities are learned, which can result into improving the entity representations of single pairs.

We additionally plot the learning curves for the different walk lengths, by randomly sampling a percentage of sentences as training data. Figure 4.6 demonstrates that, generally, all models benefit from more training data. It appears that $L = 1$ shows the slowest ascent, indicating limited capabilities compared to the walk-based models.

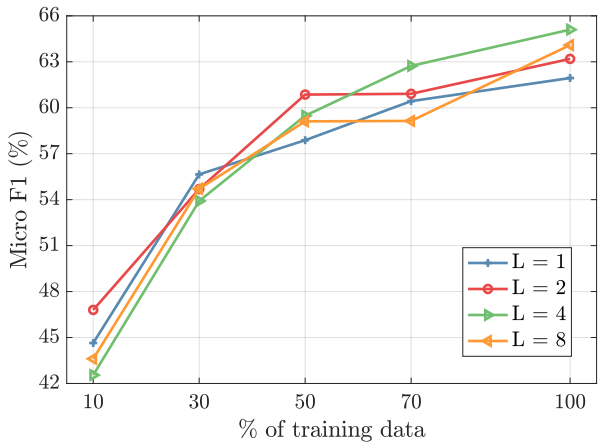


Figure 4.6: Learning curves for different walk lengths on the ACE05-D development set.

Finally, we investigate the effect of the beta (β) weight in the walk-based mechanism. For this purpose, we train our model with multiple β values ranging from 0 to 1. A beta value equal to zero indicates that only extended walk representations (2-hops or more) are used to construct the final pair representation. On the other hand, a beta value equal to one indicates that only short edge representations will be used for the construction of the pair representation, without considering longer ones, as results from Equation (4.9).

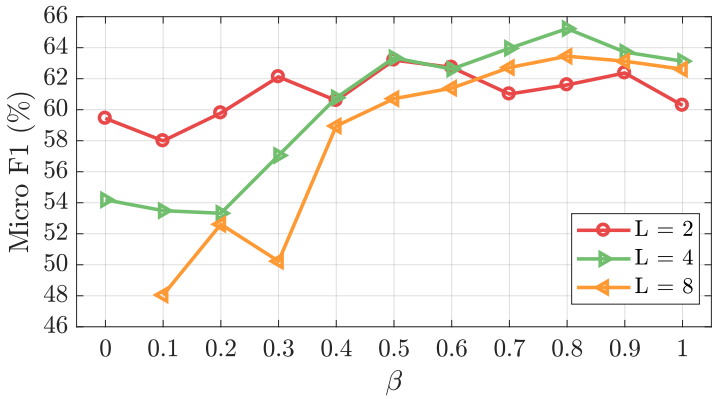


Figure 4.7: Performance as a function of different β values for multiple walks length on the ACE05-D development set. The performance of $\beta = 0$ for $L = 8$ is not reported as it is below 20%.

As we can see from Figure 4.7, the increase of the beta value leads to better performance for longer walks, with peaks around $\beta = 0.8$. However, for a single-hop ($L = 2$), the performance follows a marginally ascending trend and ranges from 58% to 63%. The best beta value for this setting is equal to 0.5, i.e. both 1-step and 2-step representations are weighed equally. This serves as an indication that shorter walks, i.e. of length 1 and 2, contain important information regarding associations between named entities. Yet, longer walks are more likely to contain noise since many pair representations are used to form the walk. As a result, we deem that, when representations of longer walks are weighed less, we can alleviate adding too much noise and simultaneously consider a small amount of potentially useful information. In addition, the fact the $L = 4$ surpasses $L = 2$ with $\beta \geq 0.7$ indicates that we indeed need information from longer inference chains, although it is crucial to control their contribution into the pair representation. This observation confirms our expectation that the shortest walk (or path) between two named entities contains the most important information for relation extraction, as proved by Borgwardt and Kriegel (2005); Xu et al. (2015c).

4.5.3 Edge representation enhancements

We also investigate the effect of the initial edge representations in the model for different walk lengths. We perform ablation analysis on the different edge representation enhancements. In particular, we begin with a baseline edge representation that consists of the concatenation of two entities using their LSTM representation. We gradually add bi-directionality, relative position, entity types and contextual embeddings.

As we can observe from Table 4.12, bi-directionality, positional and semantic entity type embeddings contribute to the improvement of the initial edge representation. Even when the initial representation is considered weak, i.e. does not contain rich contextual information, the walk-based model improves over $L = 1$ despite the low informativeness of the representations. In particular, the semantic entity type embeddings improve the performance for a large margin over all models. This is to be expected, as the information of the type of a pair's arguments are a strong inductive bias for the relation that the two entities share.

	Micro F1 (%)			
Setting	$L = 1$	$L = 2$	$L = 4$	$L = 8$
Baseline LSTM	50.13	49.16	52.53	50.55
+ Bi-direction	54.61	53.84	56.51	55.42
+ position	56.28	56.46	58.81	57.61
+ entity type	64.16	64.03	65.54	64.04
+ context words	62.86	62.74	65.30	64.63
+ context entities	63.23	62.43	65.04	64.73
+ context words, entities	61.94*	63.19	65.10	64.09

Table 4.12: Ablation analysis in the ACE05-D development set for different model enhancements. $*$ indicates significance at $p < 0.05$ with the last model.

However, it appears that additional pair-specific context information into the initial edge representations hurts the performance, mostly for no walks ($L = 1$) or short walks ($L = 2$). We experimented with adding word or entity context separately, which appear to have a similar effect. The decrease in the performance can be attributed to several reasons. Firstly, the baseline model has limited expressive power and additional parameters can lead to overfitting. Secondly, the BiLSTM layer already contains enough contextual information into the word representations and, consequently, the entity representations. Thirdly, we can observe that the performance of $L = 4$ and $L = 8$ does not drop so much with additional context. A possible explanation might be that, for longer walks, the original edge information fades with time, since we aggregate walks in an iterative manner. As a result, by adding context, we can enhance the pair representation for longer walks. Overall, we conclude that, despite our intuition that additional context might be required to construct unique edge representations for each pair, it is not indispensable, since both the linear reduction layer in Equation (5.8) and the walk-based mechanism (Equations (4.8)-(4.9)) produce unique edge representations.

We finally analyse the attention mechanism that we used to construct pair-specific

context representations. At this point, we should note that there has been intense debate, recently, about whether attention mechanisms are interpretable and if they should be used to explain decisions made by the models. Recent work has shed some light on possible reasons that attention weights might be misdealing, or that they can indeed provide some useful insights of the model's behaviour (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019). Here, we do not aim to use the attention weights in order to assess the correctness or quality of the model's predictions; rather, we aim to observe what associations can be captured when using this attention mechanism inside the model.

In Table 4.13, we present some examples where our best performing model ($L = 4$) succeeds or fails to detect the relation between the pair, along with the attention weights produced for the sentence where the pair belongs.

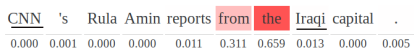
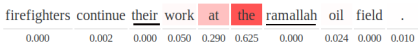
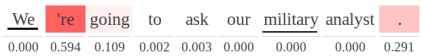
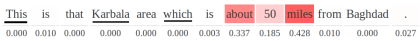
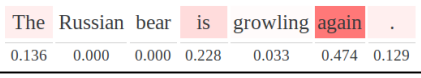
Pred.	Truth	Arg1	Arg2	Sentence
PHYS	PHYS	Rula Amin	capital	
PHYS	PHYS	firefighters	field	
ORG-AFF	ORG-AFF	analyst	our	
PHYS	NR	area	Baghdad	
PHYS	NR	He	Greece	
ORG-AFF	NR	Russian	bear	

Table 4.13: Attention heatmaps for the $L = 4$ walk-based model on the ACE05-D development set. The underlined words correspond to additional named entities in the sentence.

For the first two examples, the attention mechanism can identify words that are important for the pair, however, with unexpected weights. The word *the* receives higher attention score compared to *from* or *at* in each example, respectively. Regarding the third example, the attention weights look arbitrary with respect to the target pair. For the next two sentences, it appears that *from* is weighed very less but other words in between the pair are weighed more. In the last example, the attention mechanism assigns different weights to several context words. This seems reasonable as there is no explicit relevant context word for this particular relation.

The behaviour of attention in our model, seems not to be generally consistent to what we would expect. One explanation regards the shared sequence layer among the pairs of a sentence. For instance, consider example number three from Table 4.13. There are four named entities in this sentence, hence six pairs. As all pairs are predicted simultaneously and also use the same input sequence representation, the attention mechanism also attempts to attend to important words for each pair at the same time. It can typically succeed when there are few named entities in the sentence, or if the patterns among pairs match. However, attending (via a single vector) to multiple words for different pairs simultaneously, might introduce noise to the learned vector. As a result, higher weights can be assigned to words that are more relevant to entities not participating in the target pair.

To support the above claims about simultaneous updates, we list some examples from the incorrectly identified pairs on the SemEval 2010 development set, which contains only two named entities per sentence with enough context information. The model almost always finds appropriate contextual words for the pair, as shown in Table 4.14. The fact that attention seems to attend to important elements for a given pair in this dataset and that the final prediction is wrong, clearly indicates a lack of addition information needed to detect the correct relation. For example, *from* is a common pattern for both *Entity-Origin* and *Cause-Effect* relations, but the named entities themselves indicate that the latter is correct. It is worth noting that the highest weighted words or entities in all of the examples (Tables 4.13, 4.14) appear in the middle of the pair. We will further investigate the impact of using different attention mechanisms in the next chapter.

Pred.	Truth	Sentence
Entity-Destination	NR	New experiences throw neurons into reverse . 0.000 0.000 0.000 0.000 1.000 0.000 0.000
Instrument-Agency	NR	A nurse helps the caregiver . 0.004 0.000 0.995 0.001 0.000 0.000
Entity-Origin	Cause-Effect	Most of the steam comes from a volcano 's magma . 0.000 0.000 0.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 0.000

Table 4.14: Attention heatmaps for the baseline model on the SemEval 2010 development set. The words in boxes indicate the target named entity pair.

4.6 Related Work

Traditionally, relation extraction approaches have incorporated a large variety of hand-crafted features to represent related entity pairs (Hermann and Blunsom, 2013; Miwa and Sasaki, 2014; Nguyen and Grishman, 2014; Gormley et al., 2015). Recent models instead employ neural network architectures and achieve state-of-the-art results without heavy feature engineering.

State-of-the-art systems have proved to achieve good performance on relation extraction using RNNs (Liu et al., 2015; Cai et al., 2016; Xu et al., 2016) and CNNs (Zeng et al., 2014; Sorokin and Gurevych, 2017). However, most approaches do not take into account the dependencies between relations in a single sentence (Miwa and Bansal, 2016; dos Santos et al., 2015b; Nguyen and Grishman, 2015) and treat each pair separately. Methods that divert from this direction, such as Miwa and Sasaki (2014); Gupta et al. (2016); Sorokin and Gurevych (2017), treat the relations prediction globally. They consider interactions between entities and relations but not between relations explicitly. Other models tackle this issue using relation reasoning over entity paths. However, these approaches perform on pre-defined Knowledge Base (KB) graphs (Neelakantan et al., 2015; Das et al., 2017; Yin et al., 2018), whereas we built entity-based graphs from sentences without prior knowledge.

Recent graph-based models incorporate Graph CNNs with dependency parsing (Zhang et al., 2018b) or Graph LSTMs for cross-sentence relation extraction (Peng et al., 2017; Song et al., 2018). These models, however, encode pairwise information into the node representations instead of the edge representations between nodes, which are unique for each pair.

The most relevant work to ours is that of Sorokin and Gurevych (2017). In their work, they try to take advantage of other pairs that exist in the same sentence with the target pair. They form the final pair representation as a weighted average of the representations of the other, namely context pairs, in the sentence. However, the main difference between the two approaches is that we aim to explicitly utilise the interactions between pairs in order to form chains of associations between named entities. Instead, previous work mostly measures the similarity of the target pair with the remaining pairs in the sentence.

4.7 Conclusion

In this chapter, we elaborated on our first hypothesis (H_1) and described a novel neural network model that takes advantage of existing named entities in a sentence to support relation extraction.

Our model constructs an entity-based graph for each sentence, with named entities placed as nodes and relations between them as edges. Each edge is associated with an initial representation that is the combination of information from both the named entities that participate in the relation, as well as the pair's context. Then, an inference mechanism combines consecutive edges iteratively in order to create chains of interactions via intermediate entities in the same sentence. These chains of interactions are formed via walk generation on the entity graph, encoding up-to L-length walks between the entities of a pair into a single pair representation. This renders our model edge-oriented, in the sense that it forms and updates unique edge representations instead of nodes, compared to existing graph-based models.

We evaluated our model on four sentence-level datasets, with multiple and single pairs per sentence. We compared the performance with the state-of-the-art models and observed comparable or superior performance for the ACE 2004 and 2005 datasets without the use of any external syntactic tools. We additionally evaluated our model on a distantly supervised dataset, where we achieved better performance compared to a model which also utilises co-existing pairs in the same sentence. Finally, we showcased that our baseline can achieve decent performance on corpora that do not include multiple entities.

We conducted in-depth analysis of the model by looking into three aspects: (i) the model classification and, in particular, the directionality of the classified pairs, (ii) the walk-based mechanism and (iii) the initial edge representations. Our analysis regarding classification revealed that one direction is enough for the particular model, while two directions tend to cause confusion during learning. Moreover, the walk-based mechanism can boost the performance in cases of multiple sentential pairs. However, this improvement is relevant, as sometimes more named entities do not necessarily require longer walks.

From the analysis on the edge representation enhancements, we concluded that the semantic entity type embeddings are the most important model parameter, while the pairwise context embeddings are the least important. The attention mechanism that we incorporated is not necessary and can actually hurt performance, since the simultaneous pair prediction cannot be effectively handled via a single vector. It is thus

possible that another attention mechanism, such as multi-head attention (Vaswani et al., 2017), can encode more effectively fine-grained contextual information to several pairs at once.

Finally, the main characteristics of the proposed approach can be summarised into four factors: (i) the encoding of dependencies between relations with a shared sequence encoding layer, (ii) the simultaneous prediction of all pairs residing in a sentence, (iii) the formulation of multiple walks between named entities (i.e. chains of interactions) as fixed dimensionality vectors, and (iv) the independence from external syntactic tools.

Chapter 5

Adaptation to the Biomedical Domain

In the previous chapter, we introduced an edge-oriented graph encoding mechanism using walks on entity graphs, which can take advantage of interactions between multiple pairs in a sentence. The proposed method was evaluated against generic domain corpora, from news and encyclopedias. In this chapter, we aim to experiment with the same sentence-level mechanism addressing our second hypothesis (H_2), i.e. to assess the effectiveness of the approach across domains and evaluate whether similar observations hold, regarding the model behaviour. We present two test cases, with respect to the biomedical domain, biomedical literature documents and clinical reports, where we highlight the corresponding challenges for each test case. This chapter serves both as an application of the previously proposed method to another domain, as well as a more detailed description of the general linguistic challenges for relation extraction on biomedical text, as a more challenging domain.

5.1 Biomedical Relation Extraction

Relation Extraction has attracted particular interest in the biomedical domain as current state-of-the-art RE methods can easily extend to relations between biomedical entities. The importance of biomedical RE is reflected on the impact that automatic extraction methods can have on the domain of medical research. The increase of medical data over the last years, especially in the form of scientific articles (MedLine), urges the need for automatic extraction methods of structured information from the literature. A few examples of such cases are discovery of associations, Personalised Medicine and Pharmacovigilance.

With regard to the first case, data extracted with automatic methods can provide

assistance to clinical researchers and medical practitioners. For instance, biomedical interactions of interest (not yet discovered) can be hidden in the vast literature. Since medical research requires time and resources (lab experiments, clinical trials, etc), automatically extracting candidate associations between medical and/or molecular entities from various published articles, can point researchers towards investigating more focused directions, thus saving time and effort. A field dedicated to discovering associations with extensive application to biomedical literature, is Literature-based Discovery, introduced in the early 1980s from Don R. Swanson (Swanson, 2008). Related methods aim to link medical concepts that are discussed and analysed in separate documents, through other interactions among them, and generate lists of hypotheses of potential associations (Henry and McInnes, 2017).

Personalised medicine (PM) (Vogenberg et al., 2010) has been a very active research field in the last decade, that targets to form patient-specific treatments based on the patient's medical history and current condition. Systems with the ability to cross-reference conditions, symptoms or treatments of multiple patients or other recorded cases, can greatly benefit methods developed for stratified medicine. Pharmacovigilance is correlated with the PM field and, as the name implies, has as principal goal the identification, assessment and prevention of Adverse Drug Reactions (ADR). Additionally, abuse of drugs or misuse of wrong prescriptions, are also areas of interest, as they can result in ADRs. As different organisms exhibit different reactions to medication, clustering information about common reactions, or reactions that appear under certain conditions, can positively impact drug safety. It is worth noting that the above methods search for extracting information across various types of documents such as literature, clinical reports, prescriptions, medical records, etc.

5.1.1 Challenges

Early research in the domain has focused on Protein-Protein Interactions (PPI) with five major datasets (AIMed, BioInfer, HPRD50, IEPA and LLL). However, these corpora are relatively small. Several differences also exist between them, mainly due to different annotation strategies (Pyysalo et al., 2008). Later on, Drug-Drug Interactions (DDIs) became particularly important due to the increase of ADRs. Several studies examine novel ways to identify DDIs and also relate to the discovery of new interactions between drugs or other substances (Liu et al., 2012; Dewi et al., 2017). Additionally, multiple shared tasks have been proposed over the years for the explicit automatic identification of ADRs (Jagannatha et al., 2019) or disease-treatment relations (Uzuner

et al., 2011).

In general, biomedical relation extraction is a much more challenging task not only compared to the generic RE from news or encyclopedias but also to other biomedical tasks such as Named Entity Recognition or Sentence Similarity. As recently shown in Peng et al. (2019) even when incorporating large amounts of external resources as source of domain-specific information, the performance of RE compared to other tasks, is still lower. There exist plenty of reasons for this phenomenon that relate to the nature of entities and relations in this domain.

An initial challenge is corpora annotation for the biomedical domain, which is an expensive procedure, as domain expertise is required. In all developed bio-related corpora, human annotators have medical training and experience in the field. By contrast, in the generic domain, typically an educated person can perform an annotation process. Furthermore, relations expressed between medical entities can vary, from broad, e.g. cause-effect, to very specific, e.g. agonist, inhibitor. The difficulty of annotation is reflected in the few semantic types of biomedical entities and semantic relation categories in existing biomedical corpora. Some of those focus on binary relation extraction, i.e. whether a pair of entities shares or does not share a relation, while in other different semantic relation categories are annotated; however, they are restricted between named entities of specific types. Moreover, these annotations tend to be fine-grained as they usually derive from biomedical ontologies. There is, thus, a tendency to merge multiple semantic relation types into more broad categories to avoid data sparsity.

Other challenges emerge with respect to language. The general linguistic challenges that were described for RE in Chapter 2 are also existent for relations of the biomedical domain. However, we highlight a few that seem to be more common to this domain. Firstly, medical entities are individually complex; thus, their interactions subsequently have a particularly complex nature. Such entities can be found in text with several different names or aliases. For instance, *Suxamethonium*, is a medication used to cause short-term paralysis during anesthesia, can be found in text as, *Suxamethonium*, *Suxamethonium chloride*, *Sch*, *succinylcholine* or even *2,2'-[(1,4-dioxobutane-1,4-diyl)bis(oxy)]bis(N,N,N-trimethylethanaminium)*, where all refer to the same substance. The variance of entity surface forms is a frequent phenomenon in biomedical text, which is what motivates methods to normalise named entities into concepts to facilitate further tasks (Leaman et al., 2013).

Another linguistic challenge is hypernymy and hyponymy, where entities are classified as belonging to specific categories. For instance, *Streptococcus Pneumoniae* is under the more general *Streptococcus* category. This is also found in the general domain, but can be a regular case for biomedical text. It can generally cause issues in RE, if systems detect a relation with an entity belonging to a more general category, but fail to detect the relation with the more specific entity. In addition, biomedical texts are highly technical, since they can either describe conducted experiments or particular treatment schemes that involve multiple rounds of medication, dosages, etc. This is related to the usage of specific vocabulary, particular verbs or expressions, such as *in vitro*, *down-regulation*, *inhibits*, and so on. Finally, scientific literature contains a fair amount of speculations (Kilicoglu and Bergler, 2008). Modality auxiliaries (e.g. *maybe*, *could*), epistemic verbs or adjectives (e.g. seem, probable, possible) can be used as evidence for the existence or not of a relation, which highly depends on the annotation scheme followed by the dataset.

Our objective in the following sections is not to study any particular linguistic phenomenon but, rather, to investigate how interactions between different pairs of biomedical entities can affect the identification of other pairs in the same sentence, when multiple pairs co-exist. We particularly experiment with inclusion and exclusion of interactions between specific entity categories and examine whether they are helpful (or not) for the identification of other pairs. Additionally, we perform similar analysis to the previous chapter, in order to prove the effectiveness of our proposed model in this domain.

5.2 Scientific Articles

Our first goal is to evaluate our previously proposed edge-oriented mechanism on extracting relations from biomedical scientific articles. MEDLINE (Medical Literature Analysis and Retrieval System Online)[a] is the largest database containing free scientific journal articles for biomedical literature covering topics from medicine, nursing, pharmacy, dentistry, veterinary medicine, health care, molecular evolution and biochemistry, currently containing around 24 million articles. MEDLINE is compiled by the United States National Library of Medicine (NLM) (DeBakey, 1991) and is easily accessed through the PubMed[b] search engine, that primarily stores information about

[a] https://www.nlm.nih.gov/bsd/pmresources.html
[b] https://www.ncbi.nlm.nih.gov/pubmed

text abstracts.

A different platform from PubMed, PubMed Central (PMC)[c], is a digital repository that stores more than five million full-text articles related to biomedical and life sciences research, spanning from the late 1700s. While PubMed is a searchable database of biomedical citations and abstracts, the full-text articles are stored elsewhere. On the contrary, PMC is a free digital archive of full articles, accessible via a web browser[d]. Enhanced metadata are created from the submitted articles to PMC, creating special document identifiers as well as medical ontologies.

5.2.1 Chemical-Protein Interactions

Over the years, experts have annotated portions of scientific biomedical articles in order to assist research in automatic identification of biomedical named entities and their interactions. Most existing work revolves around interactions of Chemicals or Genes and genetics separately, where several ontologies have been developed, such as DrugBank (Wishart et al., 2007), KeGG (Kanehisa and Goto, 2000), PharmGKB (Thorn et al., 2013), ChEMBL (Gaulton et al., 2011), ChemProt KB (Taboureau et al., 2010), CTD Mattingly et al. (2006), etc. However, these databases provide general interactions between chemical compounds, genes and diseases, while they are simultaneously difficult to construct, as they require resources via experimentation or long hours of reviewing biomedical literature texts.

Group	*Eval.*	*CHEMPROT relations belonging to this group*
CPR:1	N	PART_OF
CPR:2	N	REGULATOR|DIRECT_REGULATOR|INDIRECT_REGULATOR
CPR:3	Y	UPREGULATOR|ACTIVATOR|INDIRECT_UPREGULATOR
CPR:4	Y	DOWNREGULATOR|INHIBITOR|INDIRECT_DOWNREGULATOR
CPR:5	Y	AGONIST|AGONIST-ACTIVATOR|AGONIST-INHIBITOR
CPR:6	Y	ANTAGONIST
CPR:7	N	MODULATOR|MODULATOR-ACTIVATOR|MODULATOR-INHIBITOR
CPR:8	N	COFACTOR
CPR:9	Y	SUBSTRATE|PRODUCT_OF|SUBSTRATE_PRODUCT_OF
CPR:10	N	NOT

Figure 5.1: ChemProt-BioCreative VI dataset relation categories. Faded lines correspond to semantic categories that are not used for evaluation.

One of the largest existing human annotated datasets for biomedical RE is ChemProt. The dataset focuses on the extraction of chemical-protein/gene interactions from the

[c] https://www.ncbi.nlm.nih.gov/pmc/about/intro/

[d] http://wayback.archive-it.org/org-350/20180312141605/https://www.nlm.nih.gov/pubs/factsheets/dif_med_pub.html

literature. The dataset was originally released as part of the BioCreative VI shared task (Krallinger et al., 2017), as investigation of interactions between these types of entities was underexplored. The ChemProt challenge organisers used the PubMed abstracts published between 2005 and 2014 as the challenge data. Named entities and their interactions were annotated by domain experts. The original dataset annotations include fine-grained types of interactions between chemical-protein pairs. In order to avoid redundant class definitions, the organisers grouped the annotations into 10 semantically related categories which share some underlying biological properties (Figure 5.1). For evaluation purposes alone, only 6 semantic categories were considered, including the no relation (NR) category.

5.2.2 Related Work

Early work on Chemical-Protein interactions started in Open Information Extraction, where Craven et al. (1999) used Machine Learning methods with weak supervision for particular semantic relations. Later on, Rindflesch et al. (2000) proposed *Edgar*, a natural language processing tool that uses a semantic knowledge base of biomedical terms to identify potential relations between drugs and genes.

Regarding the ChemProt BioCreative VI corpus, most proposed approaches were initially developed as part of the challenge. The best performing system was that of Peng et al. (2018) where the authors proposed an ensemble of CNN, Recurrent NN and SVM classifiers. Corbett and Boyle (2018) introduced a transfer-learning framework in combination with domain-specific, highly contextualised word embeddings. Information from different domains proved to be effective and impacted the performance. More recently, Lung et al. (2019) proposed a feature-based model with different types of lexical and semantic features; achieving, however, lower performance compared to the neural-based models. Mehryary et al. (2017) proposed two systems: an SVM classifier with hand-crafted features and an ensemble of LSTM networks that utilise the shortest dependency path between two target entities. The combination of both systems yielded competitive performance, which was later improved by using an ensemble with multiple RNN networks (Mehryary et al., 2018).

Several methods used attention mechanisms to improve relation extraction between chemicals and proteins. In particular, Liu et al. (2017b) introduced an attention RNN model and, later on, they improved their method by augmenting the training set (Liu et al., 2018). However, deviating from previous works, the authors did not mask the target named entities, i.e. did not replace the names of the chemical or gene/protein with

a unique identifier. This inevitably resulted in less informative word representations, as the number of out-of-vocabulary words increased. Following the same direction, Verga et al. (2018) proposed a Transformer-based network that was able to perform better than Liu et al. (2017b) without named entity masking. Lim and Kang (2018) incorporated a special position-feature scheme along with a Tree-LSTM, achieving similar performance with the best model during the challenge. Recently, Zhang et al. (2019b) achieved state-of-the-art performance with ELMo (Peters et al., 2018), which are highly contextualised word embeddings, and through incorporating a multi-head attention mechanism on top of a BiLSTM network.

5.2.3 Proposed Approach

We aim to apply our previously introduced model (Chapter 4) to the biomedical domain for scientific articles. In a similar fashion, the architecture consists of five layers: an embedding layer, a sentence encoding layer, an edge representation/graph construction layer, an inference layer and, lastly, a classification layer. The main differences with our previous model are featured only in the edge representation layer and the classification layer, as we describe below.

Edge Representation Layer

In our previous chapter, we discovered that the attention mechanism that we used in order to represent the context of a target named entity pair did not perform as expected. We attributed this behaviour to the simultaneous updates of the attention parameters for multiple pairs in the same sentence, that disabled learning of informative patterns for each pair separately. In order to further investigate this for the biomedical domain, we additionally experiment with a different attention mechanism, motivated by Vaswani et al. (2017). The proposed mechanism is independent of any learning parameters and is pair dependent, hence we expect to resolve the problem of identifying relation patterns for all pairs in the sentence simultaneously. In essence, this attention mechanism aims to measure the importance of an entity towards other words in the sentence. We adapt this into an argument-based framework, similar to Wang et al. (2016), by measuring the importance of each context word or entity to each pair argument.

The representation of a target named entity e_i is similar to the one in our previous architecture, with a small difference; the representation is constructed as the concatenation of the BiLSTM word representation $\mathbf{x}_{e_i}$, the representation of its semantic type

$\mathbf{t}_{e_i}$ and the relative positions to each target entity $\mathbf{p}_{e_i,e_i}$ and $\mathbf{p}_{e_i,e_j}$,

$$\mathbf{v}_{e_i} = [\mathbf{x}_{e_i}; \mathbf{t}_{e_i}; \mathbf{p}_{e_i,e_i}; \mathbf{p}_{e_i,e_j}], \tag{5.1}$$

$$\mathbf{v}_{e_j} = [\mathbf{x}_{e_j}; \mathbf{t}_{e_j}; \mathbf{p}_{e_j,e_i}; \mathbf{p}_{e_j,e_j}] \tag{5.2}$$

Here, we add the relative position to the target entity itself (something we avoided before), which is always zero, in order to have the same dimensionalities for the argument representation and the pair context representation. The purpose of this choice is to facilitate the computation of attention as described below.

The context of a given pair again consists of the remaining words, that are not part of any entity name, and named entities in the sentence, which are not part of either of the target named entities. This can be seen as a fourth-order tensor $\mathbf{C}$ where the rows correspond to the first argument, the columns correspond to the second argument and the depth corresponds to the words and named entities that belong to the context.

$$\mathbf{C}_{e_i,e_j} = \begin{bmatrix} \mathbf{x}_{e_z}; \mathbf{t}_{e_z}; \mathbf{p}_{e_z,e_i}; \mathbf{p}_{e_z,e_j} \\ \dots \\ \mathbf{x}_{w_z}; \mathbf{t}_{w_z}; \mathbf{p}_{w_z,e_i}; \mathbf{p}_{w_z,e_j} \end{bmatrix} \tag{5.3}$$

The scale-dot attention creates two attention distributions, one for each target entity, that contain weights corresponding to the significance of each context word or entity towards the target argument. As a result, two context representations are formed as the weighted average of the words in the context. The two representations are then concatenated to form the final context representation for the target pair.

The application of the scale-dot attention mechanism (Vaswani et al., 2017), is as follows, with $k \in \{e_i, e_j\}$,

$$\mathbf{a}_{k,i} = \frac{\mathbf{v}_k^\top \mathbf{z}_i}{\sqrt{d}}, \qquad \mathbf{v}_k, \mathbf{z}_i \in \mathbb{R}^d \tag{5.4}$$

$$\alpha_k = \underset{i}{\text{softmax}}\,(\mathbf{a}_k), \tag{5.5}$$

$$\mathbf{c}_k = \mathbf{C}_{e_i,e_j}\, \alpha_k^\top, \tag{5.6}$$

$$\mathbf{c}_{e_i,e_j} = [\mathbf{c}_{e_i}; \mathbf{c}_{e_j}], \tag{5.7}$$

where $\mathbf{v}_k$ is the vector representation of argument k, $\mathbf{z}_i$ is the vector representation of a context word or entity, α_k corresponds to the attention weight vector for argument k across all context words and entities, and $\mathbf{c}_{e_i,e_j} \in \mathbb{R}^{2d}$ is the final context representation

for the target pair, as the concatenation of the context representations of each argument.

Since the described attention mechanism does not depend on learned parameters, but rather measures some similarity between the target entity and the words/entities of the context, we expect it to highlight important words for each of the arguments that could further enhance the edge representations. Dimensionality reduction is performed on the resulted representation, before feeding it to the walk-based layer,

$$\mathbf{v}_{e_i,e_j}^{(1)} = \mathbf{W}_s \left[\mathbf{v}_{e_i}; \mathbf{v}_{e_j}; \mathbf{c}_{e_i,e_j}\right] \in \mathbb{R}^{d_o}, \tag{5.8}$$

where $\mathbf{W}_s \in \mathbb{R}^{d_o \times 4d}$ with $d_o < d$ and d_o is the dimensionality after reduction.

Classification Layer

According to the task annotation guidelines, the ChemProt relations were directed, i.e. only relations from a chemical to a gene/protein were annotated, and not vice versa. Hence, we choose to classify only Chemical-Protein candidates for this particular task, using a softmax classifier. We treat this task as a multi-class problem, since each pair can be assigned a particular semantic relation category.

5.2.4 Experimental Settings

We apply our previously introduced edge-oriented model on the ChemProt dataset. The dataset was pre-processed with the GENIA sentence splitter and tagger for sentence splitting and tokenisation, respectively. In case a named entity had two semantic entity types at the same time (e.g. both *Chemical* and *Gene/Protein*), we kept them as two separate instances. We removed duplicate pairs, as well as pairs that had conflicted types with other pairs (e.g. Relation A and Relation B at the same time), since their occurrences were minor. The overall statistics of the dataset after pre-processing can be found in Table 5.1.

We tuned our proposed model on the development set using randomly initialised word embeddings, without additional context information into the edge representations. We used the RoBo toolkit (Klein et al., 2017) to select the best hyper-parameters. Readers can refer to Appendix A.2 for detailed information regarding the hyper-parameter values for this dataset. We then further experimented using other types of pre-trained word embeddings and incorporated additional context into the edge representations. In all experiments, we only used the training set to train the model.

	Train	Dev.	Test
Sentences	3,771	2,203	3,074
Pairs	4,147	2,412	3,444
CPR-3	768	550	665
CPR-4	2,251	1,092	1,661
CPR-5	173	116	194
CPR-6	235	197	281
CPR-9	720	457	643
Negative pairs (%)	77.03	78.66	78
Entities	16,069	9,561	13,372
Chemical	7,891	4,717	6,741
Protein	8,178	4,844	6,631
Average sentence length	34.42	34.55	34.48
Average entities/sentence	4.26	4.34	4.35
Duplicate pairs	15	11	11
Conflicting pairs	10	4	14

Table 5.1: Statistics of the ChemProt BioCreative VI dataset.

In comparison with previous work, most existing approaches propose to *mask* the target named entities by replacing their entire span with a unique identifier (Peng et al., 2018; Corbett and Boyle, 2018; Lim and Kang, 2018). The motivation behind this technique is to reduce the number of unknown words in the pre-trained word embeddings in order to enhance the generalisability of the model. However, these models consider a single named entity pair per sentence. Since our model considers multiple named entities in a sentence at the same time, we cannot apply the masking criterion, as conflicts arise in cases where named entities share some words. We additionally want to enable the model to learn from the multiple surface forms of entities.

For this reason, we primarily compare with existing approaches that do not perform entity masking, such as Liu et al. (2017a) and Verga et al. (2018). We also compare with a few of the current state-of-the-art approaches on this dataset, despite the fact that they incorporate domain-dependent tools, additional data or named entity masking.

5.2.5 Results and Analysis

We first find the best setting for our model by experimenting with different types of context construction and walk-lengths on the development set. Randomly initialised word embeddings were used for this comparison. In more detail, we experiment with models that do not contain any additional context information into the edge representations, namely the *NoCntx* setting. We then further experiment with our previously

proposed attention mechanism, namely *Vector*, as introduced in Chapter 4, Equation (4.6). Finally, we test the proposed *Scale-Dot* attention as described in Equations (5.4)-(5.7). From all these settings, only the *Vector* attention mechanism requires a vector to be learned. The parameters are tuned for each of these settings.

Model	Micro F1 (%)		
	NoCntx	Vector	Scale-Dot
$L = 1$	52.77	52.18	52.17
$L = 2$	54.81*	53.72	54.31*
$L = 4$	52.83	53.36	54.74$^{*\diamond}$
$L = 8$	53.90	54.16*	55.68$^{*\diamond}$

Table 5.2: Performance comparison between different walk lengths and context construction techniques on the ChemProt development set. $*$ and $\diamond$ indicate significance at $p < 0.05$ in comparison with $L = 1$ and *NoCntx*, respectively.

As we can observe in Table 5.2, the baseline model ($L = 1$) performs lower than the walk-based model for all context construction mechanisms. It is not always significantly lower, however. Significantly higher performance is observed for all walk lengths when using the scale-dot attention mechanism. In general, the scale-dot attention performs better for longer walks ($L = 4$, $L = 8$), while the *NoCntx* setting is better for shorter walks ($L = 2$). A comparison of the context construction mechanisms for each walk length, shows that only the scale-dot attention yields better results than no context at all. This is aligned with our observation in the general domain, where the vector attention mechanism performed similarly to the setting without context. We speculate that additional context information in the edge representations is not always necessary. However, it can be beneficial for longer walks, depending on the method that we use to construct the edge. Overall, we believe that, ideally, a perfect initial representation for each pair (edge) will further help the walk-based layer. By further comparing the vector and the scale-dot mechanisms, they are significantly different only for $L = 8$.

We additionally illustrate in Table 5.3, heatmaps of attention weights examples for the vector and scale-dot attention mechanisms. We follow the same intuition as in the previous chapter, that attention can provide some insights on what information the network weighs more. We do not judge the performance of the model or its final decisions for each pair exclusively based on these weights. In fact, for all the examples described below, the model produces correct predictions. In the example sentence of the first block, we can see that the vector attention weighs words that are not so

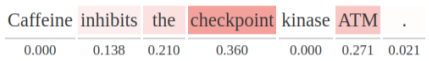
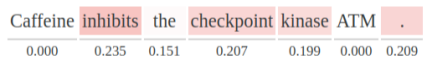
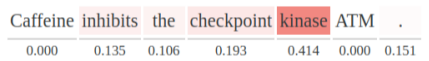
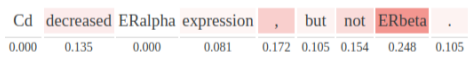
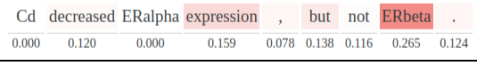
Attention	Arg1	Arg2	Sentence
Vector	Caffeine	kinase	
Scale-Dot (Arg1)	Caffeine	kinase	
Scale-Dot (Arg2)	Caffeine	kinase	
Vector	Caffeine	ATM	
Scale-Dot (Arg1)	Caffeine	ATM	
Scale-Dot (Arg2)	Caffeine	ATM	
Vector	Cd	ERalpha	
Scale-Dot (Arg1)	Cd	ERalpha	
Scale-Dot (Arg2)	Cd	ERalpha	

Table 5.3: Attention heatmaps for the $L = 8$ model on the ChemProt development set.

important for the pair, e.g. *checkpoint*, while we would expect *inhibits*. The scale-dot attention on the other hand, gives higher weight to the verb *inhibits* and *ATM* that, indeed, determine the relation. It is interesting to note that, in the second block where the second argument is ATM, scale-dot identifies the word *kinase* as the most important, as expected. The vector attention correctly highlights *inhibits*, but misses *kinase*. In the final block, the vector attention can identify the important word with a very high score, whereas scale-dot gives higher weights to other entities such as *ERbeta*. We can conclude weighing certain words more, cannot necessarily determine if the context representation is informative or not. The main difference between the two mechanisms is that scale-dot tends to pay attention to the words close to each argument which, indeed, are relation-indicative words for the most part, while vector observes the words globally. The latter is something that might not always capture what ideally we expect it to. It is worth noting that the assigned weights highly depend on the position embeddings of the context words to the target pair, which might also explain the tendency to pay attention to words with a minimum distance from each argument.

We compare our best performing model ($L = 8$) with the current state-of-the-art approaches on the ChemProt BioCreative VI test set by additionally experimenting with other types of pre-trained word embeddings. We distinguish between approaches that make use of additional training data and apply named entity masking. Our proposed

Model	Embed.	Data/Tools	Mask	P (%)	R (%)	F1 (%)
CNN (Liu et al., 2017b)	Glove	×	×	47.7	43.7	45.6
Att-GRU	Glove	×	×	48.4	49.1	48.8
Transformer (Verga et al., 2018)	Random	×	×	48.0	54.1	50.8
Hybrid (Peng et al., 2018)	PubMed+Glove	✓	✓	72.6	57.3	64.1
LSTM (Corbett and Boyle, 2018)	Glove	✓	✓	56.1	67.8	61.4
SPINN (Lim and Kang, 2018)	PubMed+PMC	✓	✓	61.5	58.9	60.2
Walks $L = 8$	Random	×	×	63.6	43.8	51.9
	Glove	×	×	64.6	44.4	52.6
	PubMed+PMC	×	×	65.1	38.5	48.3
Walks $L = 8$ (freeze)	Random	×	×	56.6	44.5	49.8
	Glove	×	×	58.3	47.8	52.5
	PubMed+PMC	×	×	62.9	50.1	55.8

Table 5.4: Performance comparison with the state-of-the-art on the ChemProt BioCreative VI test set in terms of micro-averaged precision (P), recall (R) and F1-score. The field mask indicates masking of named entities with unique identifiers.

model performs better than the models that do not incorporate such information. It is also important to note that our system has fairly low recall compared to precision while, in other models, the opposite case is observed. Our recall is similar to that of Liu et al. (2017b), who does not report training of the decision threshold for accepting a pair as related. In fact, Verga et al. (2018) trained the decision threshold for each relation category, which can improve recall. They additionally trained the model by using the same number of positive and negative instances in each batch. Since, however, the dataset is originally imbalanced, this procedure will lead to either repetition of positive instances, or ignorance of negative instances. As a result, the comparison is relatively unfair based on the fact that the model might see more instances of a certain relation, thus learn it better. Lastly, the introduction of named entity masking and pre-trained word embeddings can further lead to improvement in recall as the number of out-of-vocabulary words decreases.

Experimentation with different pre-trained word embeddings is performed by both updating the embedding layer or freezing it, i.e. disabling updates of the word embeddings during training. As we can observe from Table 5.4, updating is required

for randomly initialised word embeddings. On the other hand, when using domain-specific word embeddings (PubMed+PMC), it appears that performance worsens. We attribute this behaviour to the fact that the embeddings are already trained on domain specific corpora and further tuning possibly distorts their representations. However, this is not the case when using general-domain embeddings, such as Glove. Freezing the embedding layer improved recall but instead sacrificed precision. An overall conclusion from these comparisons is that domain-knowledge is required to achieve good performance in the biomedical domain. For the following analysis, we keep the setting with randomly initialised word embeddings as the model hyper-parameters were tuned for this setting.

Based on the confusion matrix of Table 5.5, the model tends to predict most pairs as not sharing a relation, as is expected due to the class imbalance. Additionally, it

true / pred	CPR:3	CPR:4	CPR:5	CPR:6	CPR:9	NR
CPR:3	189	80	3	1	1	391
CPR:4	31	956	2	0	11	661
CPR:5	1	12	75	6	0	100
CPR:6	0	9	6	140	0	126
CPR:9	4	26	0	0	225	463
NR	118	403	27	34	86	

Table 5.5: Confusion matrix for the $L = 8$ model on the ChemProt test set.

appears that CPR:3 and CPR:4 are often confused (this is something also observed by prior work (Liu et al., 2018; Corbett and Boyle, 2018)), while CPR:9 is some times miss-classified as CPR:4. We qualitatively report some of these cases in Table 5.6 to provide some inside regarding these errors.

Firstly, we observe that confusion between the two classes is mostly because of the presence of negative words (e.g. suppressed) along with positive ones, related to different pairs. In the first example, *icilin* activates *TPRM8* so the up-regulator (CPR:3) class is correct. However, the presence of *suppressed* which relates to *low pH* and has a negative meaning is associated with the down-regulator class (CPR:4), producing a wrong prediction. Additionally, negation is not handled properly by the model in the last example, which again causes miss-classification. The next two examples refer to a case of mishandling negation, either explicitly with *neither*, *nor* or implicitly using the verb *blocked*. In the last example, it appears that the model does not recognise that the word *substrates* expresses a product-of relation. It is worth noting that, when incorporating domain-specific embeddings, this error is averted, as domain-knowledge encodes the necessary meaning to these words.

Error	Pred.	Truth	Sentence
Context misconception	CPR:4	CPR:3	In contrast , *menthol* - and **icilin** - activated **TRPM8** currents were suppressed by low pH .
Negation	CPR:4	CPR:3	Neither **ryanodine** nor *EGTA* inhibited down - regulation of **alpha - AR** mRNA by NE .
Negation	CPR:3	CPR:4	Further , **AICAR** pretreatment blocked **PAR - 1** - induced increase in permeability of mouse - lung microvessels .
Missing semantics	CPR:4	CPR:9	**Amezinium** and debrisoquine are substrates of **uptake1** and potent inhibitors of monoamine oxidase in perfused lungs of rats .

Table 5.6: Examples of wrong predictions by the proposed model on the ChemProt development set. The named entities in **bold** indicate the target pair arguments. Words in *italics* indicate additional entities in the sentence.

Category	F1 (%)				
	$L = 1$	$L = 2$	$L = 4$	$L = 8$	Transformer
CPR:3	34.79	41.10	38.61	41.54	36.91
CPR:4	64.39	66.24	65.51	65.98	60.27
CPR:5	41.57	35.67	42.05	38.42	42.51
CPR:6	59.41	60.87	64.55	64.69	60.05
CPR:9	36.06	34.39	36.36	39.23	47.17
Micro	52.77	54.81	54.74	55.68	52.81
Macro	48.76	49.88	51.22	51.30	50.08

Table 5.7: Category-wise performance on the ChemProt development set.

Analysis on the performance for each relation category reveals that different walk lengths work better for different relation categories, similarly to the general domain. Moreover, the walk-based mechanism should be adapted for each pair or potential relation category. We compare with the transformer-based model proposed by Verga et al. (2018) using their best pre-trained model. The walk-based model outperforms CPR:3, CPR:4 (most frequent class) and CPR:6 relation categories. However, the Transformer is better for the remaining two categories.

We then repeat our entity-based analysis by measuring the performance as a function of the number of named entities in the sentence, as illustrated in Figure 5.2. Once again, we observe that the walk-based mechanism outperforms both Transformer and the baseline for single pairs per sentence. Larger improvement is observed for 3 or 4-5 entities while, for more than 10 entities, the Transformer performs slightly better.

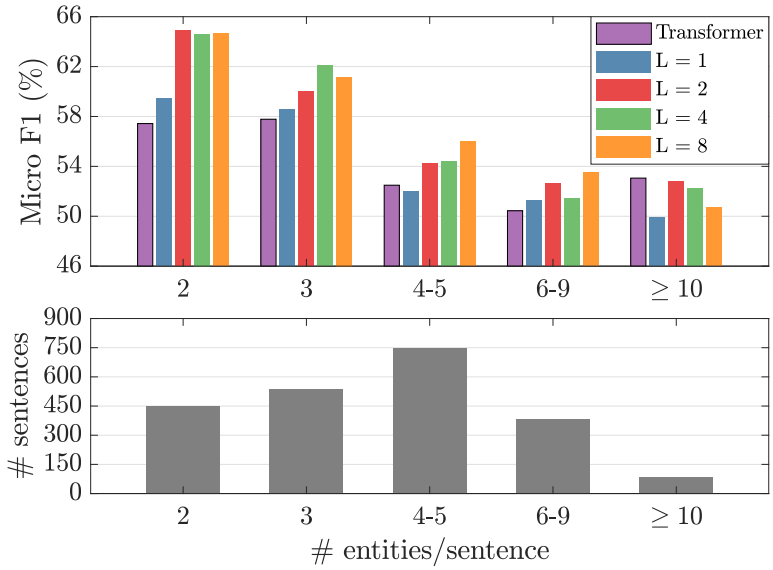


Figure 5.2: Performance on the ChemProt BioCreative VI development set as a function of the number of entities per sentence.

Since there are not many annotated entities of different semantic types in the biomedical domain, we can experiment with the contribution of different interactions in the sentence. In particular, we choose to ignore specific combinations of named entities in our walk-based mechanism. The possible combinations of named entities are re-

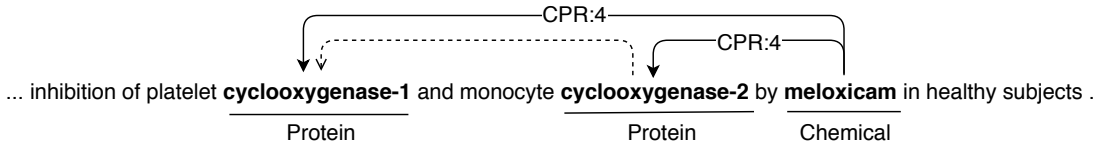


Figure 5.3: Example of ChemProt relations.

stricted to *Chemical-Protein*, *Chemical-Chemical* and *Protein-Protein*. We remove *Chemical-Chemical* and *Protein-Protein* connections from our fully connected graph. This results in disabling inference through these links. For instance, as shown in Figure 5.3, in order to generate a Chemical-Protein interaction via an intermediate Protein, we need to enable CP and PP interactions. If we disable PP interactions and there is no other Chemical entity in the sentence, we cannot represent the pair via multi-hops. We observe that, for all walk lengths, the removal of additional interactions significantly reduces performance. In some cases, the removal of particular interactions might result in improvements, such as $L = 2$. It is also notable that different relation categories are influenced by these changes if one observes the macro-F1 score. We can conclude from both the micro and macro-averaged scores that additional interactions between

Interactions	Micro F1 (%)			Macro F1 (%)		
	$L=2$	$L=4$	$L=8$	$L=2$	$L=4$	$L=8$
ALL	54.81	54.74	55.68	49.88	51.22	51.30
– CC	55.70	53.57	53.12∗	49.25	48.78	49.53
– PP	53.27∗	53.78	54.36∗	50.27	48.86	50.52
– CC, PP	51.83∗	52.52∗	53.29∗	47.51	47.55	49.25

Table 5.8: Ablation analysis for different types of interactions on the ChemProt BioCreative VI development set. CC, PP correspond to *Chemical-Protein*, *Chemical-Chemical* and *Protein-Protein* interactions. $*$ indicates significance at $p < 0.05$ with the ALL setting.

named entities in a sentence are beneficial for the detection of other pairs by providing additional context. Our model is able to take advantage of these interactions in an effective manner, although further improvements can be applied to specialise multi-hops for each pair and/or relation category.

5.3 Electronic Health Records

We now move on to investigate RE in another type of biomedical text. The current largest database storing Electronic Health Records (EHRs) is MIMIC-III (Medical Information Mart for Intensive Care) (Johnson et al., 2016). It is a freely available collection of de-identified health-related data that are associated with more than 40,000 patients who stayed in critical care units in the Beth Israel Deaconess Medical Center between 2001 and 2012. The database includes information such as demographics, vital sign measurements made at the bedside (approximately 1 data point per hour), laboratory test results, procedures, medications, caregiver notes, imaging reports, and mortality (both in and out of hospital). The language of EHRs is typically less formal compared to that of scientific literature. The records contain prescriptions or other types of information which, most of the time, are not present in full context. As a result, relation extraction methods require methods that can infer interactions with limited context. We evaluate our model on data from such a source.

5.3.1 Drug-Medication and ADE Interactions

The interactions between drugs and medication-related entities are crucial to avoid harmful consequences of pharmaceuticals. In particular, adverse drug events (ADEs)

reflect how much certain drugs can affect patients by causing undesirable side effects (Bates et al., 1995). ADEs are different from ADRs in that they refer to general adverse events that occur in various dosages, not only for normal doses to a man.

Clinical narratives and electronic health records constitute a rich source for ADE evidence. Hence, careful examination of clinical narratives can provide helpful information for pharmacovigilance. However, the large amount of EHRs, as well as their informal and unstructured nature, makes the mining of interesting interactions related to ADEs a challenging task for clinicians. To tackle this issue, NLP techniques have been widely applied on EHRs to automatically extract ADE-related information using RE methods.

We evaluated our Relation Extraction system on EHRs through participation to the 2018 n2c2 shared task on Challenges in Natural Language Processing for Clinical Data[e]. The challenge aimed to extract and classify drug-related interactions in EHRs. In particular, given an EHR with annotated drug and medication entities, the task requires the identification of potential interactions between them and their corresponding relation types. Based on the annotation scheme, the relation type between two entities can be formed as a combination of their semantic types. For instance, the relation between a Drug and a Dosage is named as a *Drug-Dosage* relation. Hence, we treat this task as a binary classification problem and simply classify an entity pair as related or not related. During the challenge, we proposed models that detect both inside sentence (intra-) and across sentence (inter-) relations and experimented with their ensembles, using gold or predicted named entities. As this was a collaborative work, this dissertation will refer to only a part of the study that regards the application of our edge-oriented, walk-based model (Christopoulou et al., 2018) to the task, for intra-sentence relations detection. More information about the entire system can be found in Christopoulou et al. (2020).

5.3.2 Related Work

Due to the lack of publicly available data, initial approaches identified potential ADEs using co-occurrence statistics and feature-based methods while evaluating on drugs with known adverse effects (Wang et al., 2009). Later, Kang et al. (2014) built a knowledge base utilising information from the Unified Medical Language System (UMLS). Drugs and ADEs were determined based on a concept matching module. The shortest

[e] https://n2c2.dbmi.hms.harvard.edu/track2

path between two concepts in the knowledge base was used to identify potential relations. Following feature-based techniques, graph topological and linguistic features were also explored to automatically detect drugs and their ADEs in unstructured text (Dasgupta et al., 2017).

Over the years, several researchers worked on creating additional annotated data with medication-drug interactions. The 2010 Informatics for Integrating Biology and the Bedside/Veteran Affairs challenge on concepts, assertions, and relations in clinical text (Uzuner et al., 2011) focused on RE among medical problem, treatment and test pairs. The best performing systems in the challenge (Roberts et al., 2010; de Bruijn et al., 2010) used dictionaries and feature-based methods, while a CNN model was proposed to achieve competitive performance a few years later (Sahu et al., 2016).

A systematically annotated corpus was generated in Gurulingappa et al. (2012b) for extraction of Drug-Dosage and Drug-ADEs relationships from medical case reports. Based on this corpus, an end-to-end system including CNN and BiLSTM networks was proposed on the shortest dependency path of an entity pair (Li et al., 2018a). The method was extended by replacing the shortest dependency path with an attention mechanism (Ramamoorthy and Murugan, 2018), achieving higher performance. ADE relation extraction was treated as a multi-label, sequence-to-sequence problem using BiLSTMs in Bekoulis et al. (2018b). Performance was further improved with adversarial training (Bekoulis et al., 2018a). Finally, Zhao et al. (2018) treated ADEs relations as event structures by proposing a two-step event extraction process, including CNNs and a beam search algorithm.

The Text Analysis Conference (TAC) Adverse Reaction Extraction from Drug Labels Track 2 (Roberts et al., 2017) asked participants to identify relations between adverse reactions and other named entities. The highest performing system in the challenge proposed a cascaded sequence labelling approach of BiLSTM conditional random fields (BLSTM-CRF) networks for end-to-end NER-RE (Xu et al., 2017) while the second ranking system used BiLSTM-attention (Dandala et al., 2017). A richer ADE-related corpus was developed by Munkhdalai et al. (2018) extending to 8 named entities and 7 relation types. They compared different models including support vector machine (SVM), LSTM and BiLSTM-attention. In the recent MADE (Medication, indication and Adverse Drug Events) 1.0 Challenge (Jagannatha et al., 2019), participants had to identify relations between medication and ADEs, indications, other signs and symptoms. Once again, BiLSTM-attention networks achieved state-of-the-art performance (Dandala et al., 2018; Li et al., 2017).

5.3.3 Motivation

In the previous section, we incorporated Chemical-Chemical and Protein-Protein interactions to enhance detection of Chemical-Protein associations. In general, interactions between particular types of entities might be unusual. For instance, in this particular task, we aim to identify interactions between Drugs and other, medication-related entities, such as Frequency, Dosage, Strength, Form. An interaction between a *Form* and a *Frequency* is not very common. On the contrary, interactions between Drugs are very frequent in biomedical text (Segura-Bedmar et al., 2013) and it has been proven that they can potentially affect the associations between drugs and ADEs (Liu et al., 2017a).

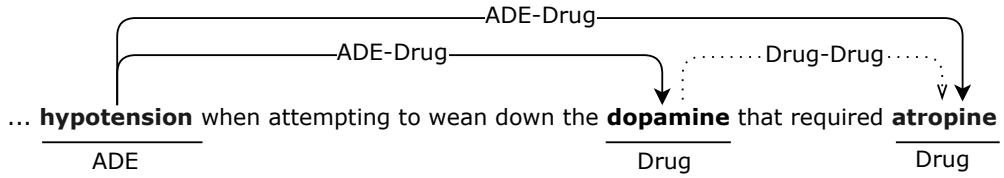


Figure 5.4: Example sentence from the n2c2 dataset with additional Drug-Drug interactions.

In the example of Figure 5.4, the direct association between *hypotension* and *atropine* is not evident when first reading the sentence. However, if we use the ADE-Drug relation *hypotension-dopamine* and the additional Drug-Drug interaction *dopamine-atropine*, the target relation *hypotension-atropine* becomes clear. In order to enable DDIs to influence other relations, but at the same time restrict the interactions between all entities in a sentence, we restrict the generated pairs to include at least one drug. Although DDIs are not annotated in the n2c2 dataset, we use them as an intermediate step to infer non–Drug-Drug relations, as in the previous task. Essentially, we infer the association between a pair using a series of interactions between entities in a sentence, including DDIs, as in the example of Figure 5.4.

5.3.4 Proposed Approach

To extract relations from EHRs, we modified the input of our model according to some observations on the dataset. Again, a major difference with existing models (Yi et al., 2017; Björne and Salakoski, 2018) is that our approach considers multiple pairs in the same sentence simultaneously.

In the first layer (i.e., the embedding layer), we map words, semantic entity types and relative positions to real-valued vectors. We follow the same approach as Zeng et al. (2014) to represent the relative position of a word to the pair of interest, which we define as the target pair. We observe that, in EHRs, several patterns express relations between entities without any supportive context words. For instance, the sentence "itraconazole$_{Drug}$ 100mg$_{Strength}$ qd$_{Frequency}$" is a typical example of a medical prescription, where no context words are present. Typically, the relations between *itraconazole* and attributes *100mg* and *qd* are inferred by humans even without explicit textual evidence. As sequences of Drug–*N* number of non-Drug entities seem important, we combine word and entity-type information as the input representation of the network, as shown in Figure 5.5. This differs from the input of the model described in Chapter 4. In preliminary experiments, we experimented with adding semantic entity type information both before and after the encoder, with the former leading to better performance. The resulting representation is then passed into a BiLSTM layer to encode sentential-context information into the word representations.

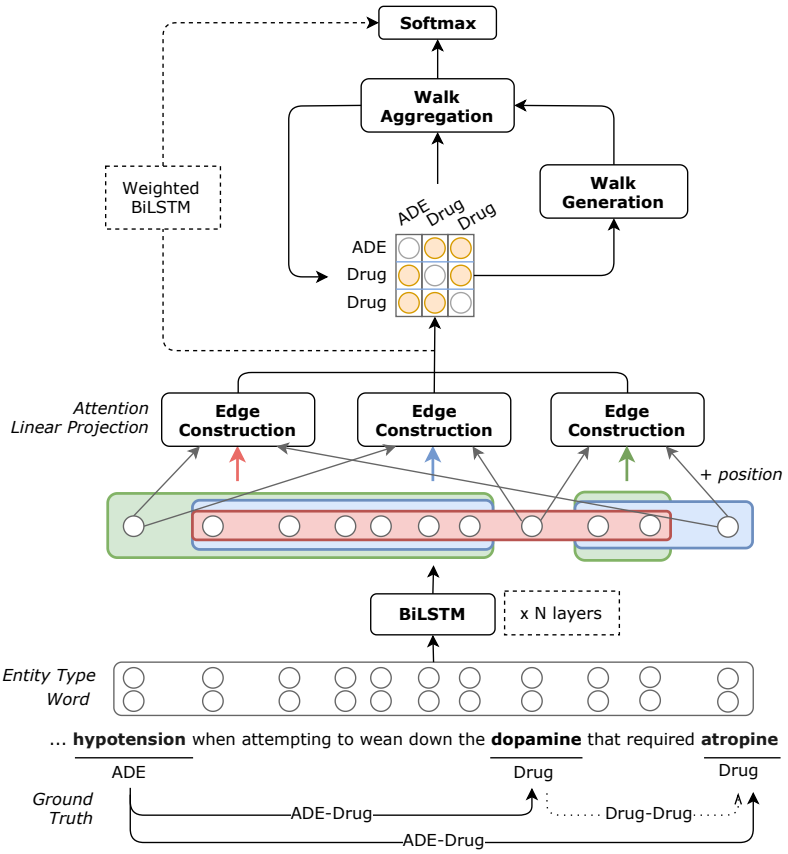


Figure 5.5: Proposed network architecture.

To enable interactions between pairs, we map a sentence into a directed graph structure, where entities constitute the nodes and edges correspond to the representation of the relation between two nodes. Figure 5.5 illustrates the proposed model, consisting of five layers. The initial edge representations of the entity graph (length $L = 1$) are equal to the entity pair representations. We combine the representations of the embedding layer and the output of the BiLSTM layer into a weighted average (highway layer (Srivastava et al., 2015)), which results in context-aware word representations. Again, this is a modification compared to the previous model (Chapter 4), which we found to perform slightly better for this task. We represent an entity by averaging its corresponding word representations. The new representations are augmented with relative position embeddings to the target entities and fed into an attention mechanism that produces entity context representation, based on the importance of the sentence words towards this entity. We employ the two-step walk-based algorithm, as described in Chapter 4. The main difference with the work on the general domain and the Chemical-Protein interactions is that we restrict the graph connections (partially connected graph) and the subsequent constructed walks between nodes. In order to form a connection, at least one of the nodes should correspond to a Drug. The final output of the walk-based layer is fed into a binary classifier to predict relation or no relation for each pair.

To summarise, the current model differs from our original model in the following aspects: (i) The input layer is the concatenation of words and semantic entity types, instead of only words, (ii) the output of the BiLSTM layer is concatenated with the word embeddings layer, instead of using only the output of the BiLSTM and (iii) a scale-dot attention is used instead of the vector attention.

5.3.5 Experimental Settings

The organisers provided 303 discharge summaries extracted from MIMIC-III (Johnson et al., 2016), annotated with Drugs, ADEs and other medication-related entities as well as their interactions. We randomly split the documents into training and development sets (80% and 20%, respectively), while duplicate relations were ignored, as shown in Table 5.9. We used LingPipe for sentence splitting and OSCAR4 for word tokenisation Jessop et al. (2011). We further split a sentence if it contained any of the following strings: "\n \n", ":\n", or "]\n". If a token contained any of the following special characters "@, ?, %,), (", we also broke it into fine-grained tokens. We additionally replaced terms that match the de-identified patient data such as "doctor X" or "patient X" with a static string of DEIDTERM, to reduce noise in the corpus.

	Training		Development	
Total Sentences		44,475		11,520
Sentences with >1 entity		7,125		1,907
Sentences with 1 entity		1,835		401
Sentences w/o entities		35,515		9,212
Sentences with 1 pair		1,672		409
	Inter	Intra	Inter	Intra
# positive relations	1,994	26,591	570	7,119
Strength-Drug	36	5,276	13	1,373
Dosage-Drug	107	3,192	33	888
Duration-Drug	29	489	4	120
Frequency-Drug	158	4,828	53	1,259
Form-Drug	123	5,060	74	1,358
Route-Drug	107	4,220	35	1,173
Reason-Drug	1,239	2,830	307	783
ADE-Drug	195	696	51	165
Negative relations (%)	97.5	59.9	97.2	56.3
Duplicate relations	19		9	
Average sentence length	21.36		21.32	
Average entities/sentence	5.39		5.4	

Table 5.9: Statistics for the n2c2 dataset for intra- and inter-sentence relations for the training and development sets.

We experimented with the following settings: (a) different walk lengths, (b) type of pre-trained word embeddings and (c) randomly removing non-related pairs in the training set, which we define as *negative instance filtering* (NIF). Significance testing was performed using the Approximate Randomisation significance test (Noreen, 1989). While training, negative filtering was used to counterbalance the bias towards the negative relation class. However, it is worth noting that this dataset contains only 60% of inter-sentence negative relations, which is significantly lower than ChemProt and the ACE corpora for the generic domain.

5.3.6 Results and Analysis

We first experimented with different walk lengths, attention mechanisms and pre-trained word embeddings. The results of various combinations are visible in Table 5.10. It appears that domain-specific word embeddings (PubMed) have generally lower performance than Randomly initialised word embeddings. These domain-specific embeddings were trained on scientific articles, which might indicate why they do not offer much in terms of performance for this corpus. Additionally, similar to our results from

the scientific articles, the scale-dot attention appears to perform slightly better compared to the vector attention, hence we chose it for our experiments. Finally, walks of length $L = 8$ give the best performance for randomly initialised word embeddings, while negative instance filtering further improves the recall of the system. This can be attributed to the fact that the n2c2 corpus contains 5 to 6 named entities per sentence, on average (Table 5.9), thus longer walks are needed to encode all interactions.

	Model	PubMed			Random		
		P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)
Vector	$L = 2$	97.15	89.90	93.39	96.56	88.79	92.51
	$L = 4$	97.46	90.27	93.73	97.76	89.90	93.67
	$L = 8$	97.51	90.29	93.76	97.81	90.51	94.02
	+ NIF	97.11	91.05	93.98	97.37	90.88	94.01
Scale-dot	$L = 2$	97.53	90.33	93.80	97.67	90.29	93.84
	$L = 4$	98.00	90.51	94.11	98.03	90.40	94.06
	$L = 8$	97.87	89.94	93.74	98.04	90.66	94.20
	+ NIF	97.13	91.05	93.99	97.34	91.15	94.14

Table 5.10: Performance of the Walk-based model on the n2c2 development set in terms of micro-averaged F1-score for different walk lengths, attention mechanisms (Vector, Scaled-dot) and pre-trained word embeddings. PubMed and Random indicate the usage of pre-trained word embeddings and randomly initialised word embeddings, respectively. NIF indicates the addition of Negative Instance Filtering.

Choosing our best setting, we compare our model with the other models that we proposed during the challenge. These are a Weighted-LSTM model that combines information from different layers into the pair representation, and a Transformer-based model, applied on single sentences. Table 5.11 reports the performance on both the development and test sets of the n2c2 dataset. As we can observe, the Walk-based model performs better than the Weighted model on the development set, but has similar performance on the test set. Despite this difference, our model has less parameters than the Weighted model, which stacks 2-BiLSTMs, as our walk-based layer consists of a single learned matrix. NIF improves the performance of the model in terms of F1-score on the test set. In general, performance of all models on this dataset is very high compared to previously evaluated datasets, due to the very high coverage in relations in each sentence.

We then investigate the contribution of the additional Drug-Drug Interactions that we allowed in the Walk-based model. For this purpose, we retrain the model without DDIs by only considering interactions between non–Drug and Drug pairs when

Model	Dev. Set			Test Set		
	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)
Weighted, 1×BiLSTM	97.02	89.85	93.30	96.32	89.13	92.58
Weighted, 2×BiLSTM	97.19	90.57	93.76	96.88	90.04	93.34
Transformer	95.49	90.46	92.91	95.79	90.11	92.86
Walks $L = 2$	97.67	90.29	93.84	-	-	-
Walks $L = 4$	98.03	90.40	94.06*	-	-	-
Walks $L = 8$	98.04	90.66	94.20*	97.45	89.46	93.28
+ Negative Filtering	97.34	91.15	94.14	96.72	90.16	93.33

Table 5.11: Performance comparison of the walk-based model with other models on the n2c2 development and test sets in terms of micro-averaged precision (P), recall (R) and F1-score (F1). $*$ indicates statistical significance at $p < 0.05$ in comparison with the Weighted model.

forming walks. In this setting, the ADE-Drug pair *hypotension-atropine* of Figure 5.4, cannot incorporate walks of $L = 2$ in its representation, as valid entity paths between the corresponding target entities cannot be formed. In essence, by removing DDIs, we restrict the valid multi-hops between two entities.

As noted in Table 5.12, the Walk-based model performs significantly lower without DDIs. Additionally, significance testing designated that different walk lengths perform similarly when excluding DDIs. A closer observation of the performance on

Category	− DDIs (%)			+ DDIs (%)		
	P	R	F1	P	R	F1
Strength-Drug	99.20	98.20	98.69	99.34	98.20	98.77
Dosage-Drug	97.65	94.58	96.09	98.53	94.58	96.52
Duration-Drug	98.32	94.35	96.30	97.52	95.16	96.33
Frequency-Drug	97.72	94.74	96.21	99.44	95.13	97.24
Form-Drug	98.55	94.30	96.38	99.05	94.23	96.58
Route-Drug	98.22	96.03	97.11	99.48	95.78	97.60
Reason-Drug	91.24	64.54	75.60	92.73	66.27	77.30
ADE-Drug	82.12	68.06	74.43	81.92	67.13	73.79
Micro	97.16	90.41	93.66	98.04	90.66	94.20*
Macro	97.31	90.13	93.45	98.03	90.32	93.89

Table 5.12: Performance comparison with inclusion (+) or exclusion (−) for Drug-Drug Interactions (DDIs) on the n2c2 development set for walks-length $L = 8$. $*$ denotes significance at $p < 0.01$ in comparison with − DDIs.

each relation category reveals that *Reason-Drug* has the largest improvement from the introduction of Drug-Drug interactions. In this particular dataset, *Reason* entities are

typically diseases or symptoms that lead to the prescription of a drug. DDIs are particularly helpful to determine the relation between a Drug and a disease or a symptom, as they can serve as cause. However, we observe that DDIs do not improve the performance of ADE-Drug associations. A possible explanation is that ADE-Drug relations are very few in this dataset, hence the model cannot learn adequate patterns for their recognition. Additionally, *ADE* entities are mostly diseases and, sometimes, a certain disease can be polysemous, i.e. both a Reason and an ADE. For instance, a Drug can cause a disease (ADE) which, in turn, requires the prescription of another Drug. This automatically renders the disease as Reason at the same time. Improving the interactions of Drugs with Reason thus might result in deterioration of ADE-Drug relations. A small improvement is also observed for Frequency-Drug relations, since DDIs can affect the frequency of drug administration.

Since we treat this task as a binary classification problem, errors are restricted to two categories. Moreover, there are no directionality errors as the relation is always from a non-Drug to a Drug entity. We try to analyse the incorrect predictions of our model using category-wise false positive rates (FPR) and false negative rate (FNR). We estimate the error rate as the proportion of all negative instances that were misclassified as positive (FPR) and the proportion of all positive instances that were misclassified as negative (FNR), as computed in Equations (5.9) and (5.10),

$$\text{FPR}_i = \frac{\text{\#FP in class i}}{\text{\#FP in class i} + \text{\#TN in class i}}, \tag{5.9}$$

$$\text{FNR}_i = \frac{\text{\#FN in class i}}{\text{\#FN in class i} + \text{\#TP in class i}} \tag{5.10}$$

Figure 5.6 visualises the false negative error rates of all intra-sentence models and their ensemble, as evaluated only on intra-sentence pairs (we do not report the FPR, as we found it was below 1% for all models and relation categories). It is observed that ADE-Drug and Reason-Drug classes have the highest probability to misclassify a pair as negative (10% for ADE and 5% for Reason). In fact, these classes are the most difficult to predict as they require well-formed context and relation-indicative words. In the sentence "*Allergies:* $Bactrim_{Drug}$ $(rash_{ADE})$", the relation between ADE and Drug is not evident as there are no keywords to support it. In contrast, Duration, Form, Strength, and other similar entities are always found close to a drug and follow a standard pattern which can be learned from sequential models. For instance,

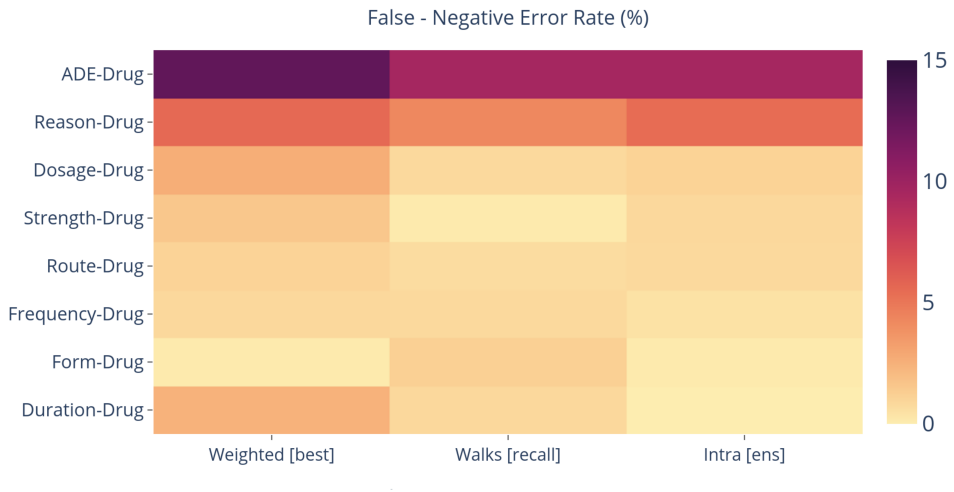


Figure 5.6: False negative error rate of intra-sentence models and their ensemble on the development set.

"Azithromycin$_{Drug}$ 250 mg$_{Strength}$ Tablet Sig$_{Form}$". Although Duration-Drug has the least positive occurrences in the dataset, our model can detect it since it is always related to the closest drug. Compared with Weighted LSTM, the Walk-based model is less biased to negative relations, as the introduction of negative filtering and the walk-inference enables the identification of more positive instances. The combination of models reduces the FNR. As we did not develop category-wise classifiers, the models try to fit all relation patterns under a single category. Additionally, since ADE- and Reason-Drug patterns are much fewer compared to other non–Drug-Drug pairs, all models tend to have lower performance on these particular categories. It is worth noting that we did not incorporate domain-specific information from external Knowledge Bases to enhance ADE-Drug detection.

We finally perform the same analysis as in previous sections to check if the walk-based mechanism benefits multi-entity sentences. As shown in Figure 5.7, performance increases with longer walks. Among different walk lengths, $L = 8$ has the best performance across multi-entity sentences, outperforming the other two models. For single pairs, the walk-based models perform similarly with the Weighted LSTM model. This is because both models consider updates from multiple pairs at the same time. This is another indication that training on multi-entity sentences, without assuming a single pair per sentence, can improve the performance on single-sentence pairs as latent, common information between pairs in the same sentence can be learned by the model. On the contrary, since the Transformer model considers only a single pair

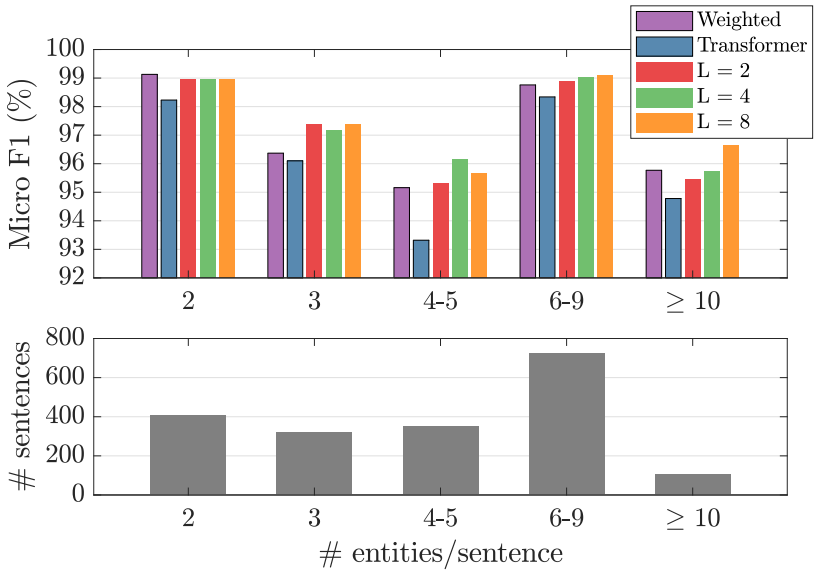


Figure 5.7: Performance of intra-sentence models on the development set on sentences with different number of entities. The bottom figure illustrates the distribution of each groups of entities.

at a time, it has 1% lower performance.

5.4 Conclusion

In this chapter we proposed to adapt the previously described edge-oriented model to the biomedical domain. Initially, we discussed some of the most common challenges associated with RE in this domain, which make detection of relations more demanding compared to the generic domain. Some challenges include high dependency on domain-knowledge, technical writing, named entities with several aliases that highlight the need for entity linking and normalisation. Other challenges also reflected on the text use cases that we investigated, scientific articles and electronic health records. For instance, both texts included technical writing, EHRs did contain less context compared to scientific articles, the latter included several nested entities.

In the first test case, we focused on Chemical-Protein interactions from the literature, were we found that the inclusion of Chemical-Chemical and Protein-Protein interactions in our graph played a significant role in performance improvement. However, incorporation of domain-specific knowledge in the form of pre-trained word embeddings can positively impact performance further, indicating the need for external knowledge in this domain. One of the largest problems of our model for this dataset

was the handling of negation, since relations appeared in text together with semantically opposite words, i.e. 'blocked the increase'. A thorough investigation of techniques that aim to focus on detecting such cases should be part of future work. We additionally compared two attention mechanisms for context construction. We confirmed our initial observation from the general domain that a learned-attention vector is not effective due to the nature of our proposed algorithm, that simultaneously treats all pairs in a sentence. By changing the mechanism, performance slightly improved, while we found that additional context information is helpful mostly for longer walks.

In the second case, we focused on Drug-Medication relations including Adverse Drug Events from EHRs. The incorporation of Drug-Drug interactions in the model assisted the prediction of Reason-Drug relations by a large margin, while it additionally improved Frequency-Drug interactions. However, ADE-Drug relations still constitute a challenging task, which is partially attributed to the polysemy of ADEs with Reason named entities as well as the need for information from domain-specific Knowledge Bases in order to increase detection.

For both test cases, we found that mapping sentences into entity-based graphs and encoding interactions among them using an edge-oriented mechanism, is beneficial for multi-entity sentences, similarly to the general domain. The mechanism is particularly helpful when relations are implicit, i.e. there is no explicit evidence in text, though they can be inferred when reading the sentence.

Chapter 6

Document-level Neural Relation Extraction

In the previous chapters, we extensively examined the case of interactions that belong in the same sentence, by proposing a sentence-level graph-based neural algorithm that performs on entity graphs. In this chapter, we propose an extension of this approach for document-level relation extraction. Our main contribution is the introduction of a simple, yet intuitive way to transform documents into graphs, without the need for syntactic tools, by addressing our third hypothesis ($H_{3.1}$). We further aim to improve both intra- and inter-sentence relation extraction using the proposed graph structure and our walk-based mechanism addressing our last hypothesis ($H_{3.2}$). The lack of document-level corpora on the generic domain for inter-sentence relation extraction forces us to evaluate the proposed approach on two biomedical domain datasets[a]. Experimental results show that we are able to achieve better performance even compared to existing approaches that incorporate additional data or tools during training, especially for the detection of inter-sentence relations. The contents of this chapter are published in Christopoulou et al. (2019).

6.1 Motivation

Although methods for extracting relations within sentences (i.e. intra-sentence relation extraction) are useful, in real-world scenarios, a large amount of relations are expressed

[a]A generic domain dataset (Yao et al., 2019) for this task became available after acceptance of this work to a peer-reviewed conference.

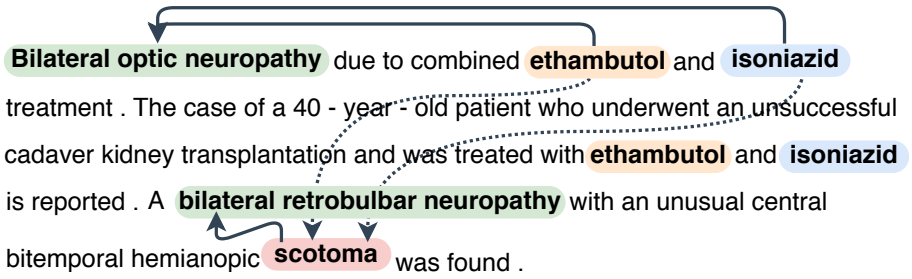


Figure 6.1: Example of document-level, inter-sentence relations adapted from the CDR dataset (Li et al., 2016a). The solid and dotted lines represent intra- and inter-sentence relations, respectively.

across sentences. The task of identifying these relations is named inter-sentence Relation Extraction. Typically, inter-sentence relations occur in textual snippets with several sentences, such as within documents. In these snippets, an entity can be repeated under the same phrase or alias, that corresponds to different *entity mentions* of the same *entity concept*. In order to identify the multiple mentions of an entity concept, a Knowledge Base is usually involved. Each named entity in the snippet is associated with a unique Knowledge Base identifier (KB ID), through a process known as Named Entity Linking (NEL) (Hachey et al., 2013). Mentions that are grounded to the same KB ID can be seen as co-referring mentions belonging to the same *entity concept*. However, these particular occurrences are restricted to nouns and proper names, in contrast to pronouns that can also be considered to co-refer with named entities in co-reference resolution tasks (Clark and González-Brenes, 2008).

We aim to use the multiple entity mention along with the entity concepts in which they belong, together with other structural elements of the document, to enhance the identification of inter-sentence relations. For instance, one can look at the example of Figure 6.1. The entities *bilateral optic neuropathy*, *ethambutol* and *isoniazid* have two mentions each (appear two times), while the entity *scotoma* has one mention (appears one time). The relation between the chemical *ethambutol* and the disease *scotoma* is clearly inter-sentential (dotted lines), since there is no occurrence of the two in the same sentence. Their association can only be determined if we consider the interactions between the mentions of these entities in different sentences. A mention of *bilateral optic neuropathy* interacts with a mention of *ethambutol* in the first sentence. Another mention of the former interacts with the mention of *scotoma* in the third sentence. This chain of interactions can help us infer that the entity *ethambutol* has a relation with the entity *scotoma*.

The most common technique currently used to handle multiple mentions of named

entities (concepts) is Multi-Instance Learning (MIL). Initially, MIL was introduced by Riedel et al. (2010) for noise reduction purposes in corpora that were created using distant supervision (Mintz et al., 2009). As mentioned in Chapter 2, MIL in this setting considers multiple sentences (bags-of-sentences), each of which contain a pair of entities. A bag-of-sentences contains all the sentences where a target named entity pair can be found. The relaxed assumption states that at least one sentence that mentions the two entities might express this relation. Verga et al. (2018) introduced another MIL setting for relation extraction between named entities in a document. In this setting, entities mapped to the same KB ID are considered to be mentions of an entity concept and pairs of mentions correspond to the pair's multiple instances. In order to construct distantly supervised corpora, relations are given to pairs of entity concepts, if they are in the same document and are known to share a relation in the KB. As a result, the relation between the two concepts can be either because they are mentioned in the same sentence, or by inter-sentence inference, or from the meaning of the entire document. It is thus necessary to model interactions that take place in the entire documents, in order to detect relations between concepts.

As mentioned in the previous chapter, however, this type of document-level RE between named entity concepts is not so common in the generic domain. Typically, generic NEs do not have so many aliases and, also, the most interesting interactions are observed when they co-occur in the same sentence (Banko et al., 2007). On the contrary, in the biomedical domain, document-level relations are particularly important given the numerous aliases that biomedical entities can have (Quirk and Poon, 2017).

To deal with document-level RE, recent approaches assume that only two mentions of the target entities reside in the document (Nguyen and Verspoor, 2018; Verga et al., 2018) or utilise different models for intra- and inter-sentence RE (Gu et al., 2016; Li et al., 2016b; Gu et al., 2017). In contrast with approaches that employ sequential models (Nguyen and Verspoor, 2018; Gu et al., 2017; Zhou et al., 2016a), graph-based neural approaches have proven useful in encoding long-distance, inter-sentential information (Peng et al., 2017; Quirk and Poon, 2017; Gupta et al., 2019). These models interpret words as nodes and connections between them as edges. They typically perform on the nodes by updating their representations during training. However, a relation between two entities depends on different contexts, especially in a document. It could thus be better expressed with an edge connection that is unique for the pair. A straightforward way to address this is to create graph-based models that rely on edge representations that rather focus on node representations, which are shared between

multiple entity pairs.

We propose to tackle document-level, intra- and inter-sentence RE using MIL, with a graph-based neural model, when entity concept annotations are available, i.e. entity linking is already performed to associate entity mentions to KB concepts. Our main objective is to infer the relation between two entities by exploiting other interactions in the document, that are more general than word-level interactions. We construct a document graph with heterogeneous types of nodes and edges to better capture different dependencies between elements of the document. In the proposed graph, a node corresponds to either entities, mentions, or sentences, instead of words. Connections between distinct nodes are derived from simple heuristic rules, while we generate different edge representations for each connection between two nodes. Differently from our previous work on sentences, we enable construction of edge representations even between nodes that were not initially connected, in order to form concept-level pair representations.

6.2 Proposed Approach

We build our model as a significant extension of our previously proposed sentence-level model, introduced in Chapter 4 (Christopoulou et al., 2018) for document-level RE. The most critical difference between the two models, is the introduction and construction of a *partially-connected* document graph instead of a fully-connected sentence-level graph. Additionally, the document graph consists of *heterogeneous* types of nodes and edges in comparison with the sentence-level graph that contains only entity-nodes and single edge types among them. Furthermore, the proposed approach utilises multi-instance learning when mention-level annotations are available. This means that the proposed approach can identify interactions both between named entity concepts, when NEL annotations are available, or named entity mentions when they are not.

Task Setting

For clarification purposes, we consider the target task as document-level Relation Extraction between concept-level named entities. The input is an annotated document. The annotations include concept-level entities (with assigned KB IDs), as well as multiple occurrences of each entity under the same phrase or alias, i.e., entity mentions, in

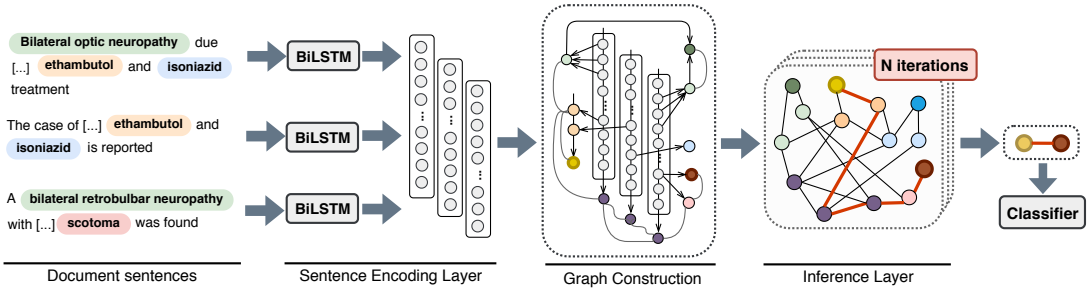


Figure 6.2: Abstract architecture of the proposed approach. The model receives a document and encodes each sentence separately. A document-level graph is constructed and fed into an iterative algorithm to generate edge representations between the target entity nodes. Some node connections are not shown for brevity.

the document. We consider the associations of mentions to concept entities as given (associated KB IDs), assuming the named entity linking is already applied on the corpus. The objective of the task is given an annotated document as such, to identify the relation (or not) for all the concept-level pairs in that document. For the remainder of the chapter, we will refer to concept-level annotations as *entities* and mention-level annotations as *mentions*, for convenience.

6.2.1 Sentence Encoding Layer

Following our previous work, first, each word in the sentences of the input document is transformed into a dense vector representation, i.e., a word embedding. The vectorised words of each sentence are then fed into a BiLSTM network, named the encoder. The output of the encoder results in contextualised representations for each word of the input sentence.

6.2.2 Graph Layer

The contextualised word representations from the encoder are used to construct a document-level graph structure, where both nodes and edges are represented by n-dimensional vectors. The graph layer comprises of two sub-layers, a node construction layer and an edge construction layer. We compose the representations of the graph nodes in the first sub-layer and the representations of the edges in the second one.

6.2.2.1 Node construction

We propose the formation of three distinct types of nodes in the graph, given the existing elements in the document: mention nodes (M) n_m, entity nodes (E) n_e, and sentence nodes (S) n_s. We adopt a simple approach, where each node representation is computed as the average of the embeddings of different elements. Firstly, mention nodes correspond to different mentions of entities in the input document. The representation of a mention node is formed as the average of the words (w) that the mention contains. Secondly, entity nodes represent unique entity concepts. The representation of an entity node is computed as the average of the mention (m) representations associated with the entity. Finally, sentence nodes correspond to sentences. A sentence node is represented as the average of the word representations in the sentence. In order to distinguish different node types in the graph, we concatenate a node type embedding to each node representation. The final node representations for each node type are then estimated as follows,

$$\mathbf{n}_{m_i} = [\underset{w \in m_i}{\text{average}}(\mathbf{w}); \mathbf{t}_m], \tag{6.1}$$

$$\mathbf{n}_{e_j} = [\underset{m \in e_j}{\text{average}}(\mathbf{m}); \mathbf{t}_e], \tag{6.2}$$

$$\mathbf{n}_{s_k} = [\underset{w \in s_k}{\text{average}}(\mathbf{w}); \mathbf{t}_s], \tag{6.3}$$

where m, e and s correspond to mentions, entities and sentences, respectively, $\mathbf{w}$ denotes a word embedding, $\mathbf{m}$ denotes a mention embedding, $\mathbf{t}$ corresponds to the node type embedding and *average* corresponds to the averaging operation.

6.2.2.2 Edge construction

We construct a non-directed adjacency matrix for the heterogeneous graph by using heuristic rules that stem from the natural associations between the elements of a document, i.e., mentions, entities and sentences. We do not directly connect entity nodes, as we aim to construct an edge representation between them using other existing edges in the graph. As a result, the entity-to-entity (EE) relations will be inferred. We define five types of edges between different nodes and further construct unique edge representations for each one, using the following criteria:

Mention-Mention (MM): Co-occurrence of mentions in a sentence might be a weak

indication of an interaction. For this reason, we create mention-to-mention edges only if the corresponding mentions reside in the same sentence.
The edge representation between each mention pair m_i and m_j is generated by concatenating the representations of the nodes $\mathbf{n}_{m_i}$ and $\mathbf{n}_{m_j}$, the contexts $\mathbf{c}_{m_i,m_j}$ and a distance embedding associated with the distance between the two mentions $\mathbf{d}_{m_i,m_j}$, in terms of intermediate words:

$$\mathbf{x}_{\text{MM}} = [\mathbf{n}_{m_i}; \mathbf{n}_{m_j}; \mathbf{c}_{m_i,m_j}; \mathbf{d}_{m_i,m_j}] \tag{6.4}$$

Here, we generate the context representation for these pairs in order to encode local, pair-centric information. We use an argument-based attention mechanism (Wang et al., 2016), to measure the importance of other words in the sentence towards the mention of interest, with $k \in \{1,2\}$ as the mention arguments,

$$\begin{aligned} \alpha_{k,i} &= \mathbf{n}_{m_k}^{\top} \mathbf{w}_i, \\ \mathrm{a}_{k,i} &= \frac{\exp(\alpha_{k,i})}{\sum_{j\in[1,n],j\notin m_k} \exp(\alpha_{k,j})}, \\ \mathrm{a}_i &= (\mathrm{a}_{1,i} + \mathrm{a}_{2,i})/2, \\ \mathbf{c}_{m_1,m_2} &= \mathbf{H}^{\intercal}\, \mathbf{a}, \end{aligned} \tag{6.5}$$

where $\mathbf{n}_{m_k}$ is a mention node representation, $\mathbf{w}_i$ is a sentence word representation, a_i is the attention weight of word i for mention pair m_1, m_2, $\mathbf{H} \in \mathbb{R}^{w\times d}$ is a sentence word representations matrix, $\mathbf{a} \in \mathbb{R}^{w}$ is the attention weights vector for the pair and $\mathbf{c}_{m_1,m_2}$ is the final context representation for the mention pair.

Mention-Sentence (MS): Mention-to-sentence nodes are connected only if the mention resides in the sentence. Their initial edge representation is constructed as a concatenation of the mention and sentence nodes,

$$\mathbf{x}_{\text{MS}} = [\mathbf{n}_m; \mathbf{n}_s] \tag{6.6}$$

Mention-Entity (ME): We connect a mention node to an entity node if the mention is associated with the entity, i.e. correspond to the same concept,

$$\mathbf{x}_{\text{ME}} = [\mathbf{n}_m; \mathbf{n}_e] \tag{6.7}$$

Sentence-Sentence (SS): Motivated by Quirk and Poon (2017), we connect sentence

nodes to encode non-local information as well discourse associations. The main difference with prior work is that our edges are unlabelled, non-directed and span multiple sentences. To encode the distance between sentences, we concatenate it to the sentence node representations in the form of an embedding:

$$\mathbf{x}_{\text{SS}} = [\mathbf{n}_{s_i}; \mathbf{n}_{s_j}; \mathbf{d}_{s_i,s_j}] \tag{6.8}$$

Since intermediate sentences might contain redundant information about the association between a pair of interest, we choose to experiment with directly connecting sentences that are not adjacent. For instance, in the example document of Figure 6.1, the second sentence is not really necessary for the identification of the inter-sentence association between *bilateral optic neuropathy* and *scotoma*. In particular, we consider $\mathbf{SS_{direct}}$ as adjacent, ordered edges (distance $= 1$) and $\mathbf{SS_{indirect}}$ as indirect, non-ordered edges (distance > 1) between S nodes, respectively. In our setting, **SS** denotes the combination of the twp as $\text{SS} = \text{SS}_{\text{direct}} \cup \text{SS}_{\text{indirect}}$.

Entity-Sentence (ES): To directly model entity-to-sentence associations and create shorter paths, we connect an entity node to a sentence node if at least one mention of the entity resides in this sentence,

$$\mathbf{x}_{\text{ES}} = [\mathbf{n}_e; \mathbf{n}_s] \tag{6.9}$$

The previously described edge representations have different dimensionalities. Hence, to result in edge representations of equal dimensionality, we use different linear reduction layers for different edge representations,

$$\mathbf{e}_z^{(1)} = \mathbf{W}_z \, \mathbf{x}_z, \tag{6.10}$$

where $\mathbf{e}_z^{(1)}$ is an edge representation of length 1, $\mathbf{W}_z \in \mathbb{R}^{d_z \times d}$ corresponds to a learned matrix and $z \in [\text{MM}, \text{MS}, \text{ME}, \text{SS}, \text{ES}]$. We expect that the linear layers will learn the best features that describe each edge type.

6.2.3 Inference Layer

We then try to directly model interactions between different pairs of nodes in the graph and consequently generate edges between entity nodes. For this purpose, we adapt our two-step inference mechanism, proposed in Christopoulou et al. (2018). In cases

where there is an existing edge in the graph, we update its representation whereas, if a new edge is created, we construct its representation using other edges in the graph.

We initialise the graph only with the edges described in Section 6.2.2.2, meaning that direct entity-to-entity (EE) edges are absent. We can only generate EE edge representations by representing a path (or walk) between their nodes.

For completeness, we describe again our two-step procedure with the corresponding notation in order to match the previous steps of the method. We first generate a walk between two nodes i and j using intermediate nodes k, by combining the representations of two consecutive edges $\mathbf{e}_{ik}$ and $\mathbf{e}_{kj}$. This action generates a new edge representation that includes walks of double length. We combine all existing walks between i and j through k. The i, j, and k nodes can be any of the three node types E, M, or S. Intermediate nodes without adjacent edges to the target nodes are ignored.

$$f\left(\mathbf{e}_{ik}^{(l)}, \mathbf{e}_{kj}^{(l)}\right) = \sigma\left(\mathbf{e}_{ik}^{(l)} \odot \left(\mathbf{W}\, \mathbf{e}_{kj}^{(l)}\right)\right), \tag{6.11}$$

where σ is the sigmoid non-linear function, $\mathbf{W} \in \mathbb{R}^{d_z \times d_z}$ is a learned parameter matrix, $\odot$ refers to element-wise multiplication, l is the length of the walk and $\mathbf{e}_{ik}$ corresponds to the representation of the edge between nodes i and k.

During the second step, we aggregate the original edge representation (short walk) and the new edge representation (longer walk) resulted from Equation (6.11) with linear interpolation, as follows:

$$\mathbf{e}_{ij}^{(2l)} = \beta\, \mathbf{e}_{ij}^{(l)} + (1-\beta) \sum_{k \neq i,j} f\left(\mathbf{e}_{ik}^{(l)}, \mathbf{e}_{kj}^{(l)}\right), \tag{6.12}$$

where $\beta \in [0,1]$ is a scalar that controls the contribution of the shorter walks. If a multi-hop representation cannot be formed in this step, we only take into account the original pair representation with $\beta = 1$.

In cases where we can construct a new edge representation, we choose a high value for β in order to give more weight to the shorter walks, as was previously discussed (Xu et al., 2015c; Borgwardt and Kriegel, 2005). This choice also results from our previous experiments in the biomedical domain, that showed that a β value around 0.8 is a good choice for walks longer than 4. After N iterations, the final edge representation will correspond to walks between different elements of length up-to 2^N.

It is important to note at this point that, in order to represent an entity-to-entity edge (EE), we need at least 2-hops for intra-sentence detection, via an E-S-E path (meaning

that at least one mention of the first entity co-occurs with a mention of the other entity). We also need 4-hops to represent an inter-sentence entity-to-entity association via an E-S-S-E path, if we allow $SS_{indirect}$ edges.

6.2.4 Classification

To classify the concept-level entity pairs of interest, we incorporate a softmax classifier using the entity-to-entity edges (EE) of the document graph, that correspond to the concept-level entity pairs. We only classify pairs based on the semantic restrictions of the dataset, i.e. each entity-concept must belong to a specific semantic category.

$$\mathbf{y} = \text{softmax}\left(\mathbf{W}_c\, \mathbf{e}_{\text{EE}} + \mathbf{b}_c\right), \tag{6.13}$$

where $\mathbf{W}_c \in \mathbb{R}^{r \times d_z}$ and $\mathbf{b}_c \in \mathbb{R}^r$ are learned parameters of the classification layer and r is the number of relation categories.

6.3 Experimental Settings

The model was developed using PyTorch[b] (Paszke et al., 2017). We incorporated early stopping to identify the best training epoch and used Adam (Kingma and Ba, 2015) as the model optimiser. A detailed version of the model hyper-parameters can be found in the Appendix A.3.

6.3.1 Data and Task Settings

We evaluated the proposed model on two datasets that belong to the biomedical domain. The first dataset is human annotated, while the second was automatically constructed using Distant Supervision.

CDR (BioCreative V). The Chemical-Disease Reactions dataset was created by Li et al. (2016a) for document-level RE. It consists of 1,500 PubMed abstracts that are split into three equally sized sets for training, development and testing. The dataset was manually annotated with binary interactions between Chemical and Disease concepts. For this dataset, we utilised PubMed pre-trained word embeddings (Chiu et al., 2016).

[b]Code for this model is available in https://github.com/fenchri/edge-oriented-graph

Due to the small size of the dataset, some approaches create a new split from the union of train and development sets (Verga et al., 2018; Zhou et al., 2018). We chose to merge the train and development sets and re-train our model in its entirety, for evaluation on the test set as Lin et al. (2016) and Zhou et al. (2016a) suggest. To compare with related work, we followed Verga et al. (2018) and Gu et al. (2016) and ignored non-related pairs that correspond to general concepts (MeSH vocabulary hypernym filtering).

GDA (DisGeNet). The Gene-Disease Associations dataset was introduced by Wu et al. (2019), containing 30,192 MEDLINE abstracts, split into 29,192 articles for training and 1,000 for testing. The dataset was automatically annotated with binary interactions between Gene and Disease concepts at the document-level, using distant supervision. Associations between concepts were generated by aligning the DisGeNet (Piñero et al., 2016) platform with PubMed abstracts. We randomly split the training set into a 80/20 percentage split as training and development sets, repectively. For the GDA dataset, we used randomly initialised word embeddings.

	Train	Dev.	Test
Documents	500	500	500
Sentences	4,621	4,626	4,847
Positive pairs	1,038	1,012	1,066
Intra	754	766	747
Inter	284	246	319
(%)	27.3	24.3	29.9
Negative pairs	4,202	4,075	4,138
Entities			
Chemical	1,467	1,507	1,434
Disease	1,965	1,864	1,988
Mentions			
Chemical	5,162	5,307	5,370
Disease	4,252	4,328	4,430
Avg. entities/doc	6.9	6.7	6.8
Avg. sentences/doc	9.2	9.3	9.7
Avg. mentions/entity	2.7	2.9	2.9
Avg. sentence length	25.6	25.4	25.7

Table 6.1: CDR dataset statistics.

	Train	Dev.	Test
Documents	23,353	5,839	1,000
Sentences	236,010	58,589	9,975
Positive pairs	36,079	8,762	1,502
Intra	30,199	7,408	1,273
Inter	5,880	1,354	229
(%)	16.2	15.4	15.2
Negative pairs	96,399	24,362	3,720
Entities			
Gene	46,151	11,406	1,903
Disease	67,257	16,703	2,778
Mentions			
Gene	205,457	51,410	8,404
Disease	226,015	56,318	9,524
Avg. entities/doc	4.9	4.8	4.7
Avg. sentences/doc	9.1	9.1	10
Avg. mentions/entity	3.8	3.8	3.8
Avg. sentence length	32.9	32.9	29.7

Table 6.2: GDA dataset statistics.

In Tables 6.1-6.2, we summarise the statistics for the CDR and GDA datasets, respectively. For all datasets, we used the GENIA Sentence Splitter[c] and GENIA Tagger[d] for sentence splitting and word tokenisation. We additionally removed mentions in the given abstracts that were not grounded to a Knowledge Base ID (ID equal to -1). It is interesting to observe that the percentage of inter-sentence pairs in the CDR dataset is around 30%, contrary to 15% in the GDA dataset.

6.3.2 Model Settings and Comparisons

We explore multiple settings of the proposed graph using different edges (MM, ME, MS, ES, SS) and edge enhancements (node type embeddings, mention-pairs context embeddings, distance embeddings). We name our model *EoG*, an abbreviation for Edge-Oriented Graph. We briefly describe the model settings and the models we compared it to. In all of the following experiments, we used the Approximate Randomisation Significance Test (Noreen, 1989) when performing comparisons.

Additional comparisons are made with three baseline models we create. EoG refers to our main model with edges {MM, ME, MS, ES, SS}. The EoG (*Full*) setting refers to a model with a *fully connected* graph, where the graph nodes are all connected to each other, including E nodes. For this purpose, we introduce an additional linear layer for the EE edges as in Equation (6.10). The EoG (*NoInf*) setting refers to a *no inference* model, where the iterative inference algorithm (Section 6.2.3) is ignored. The concatenation of the entity node embeddings is used to represent the target pair. In this case, we also make use of an additional EE linear layer for EE edges, since they cannot be constructed otherwise. Finally, the EoG (*Sent*) setting refers to a model that was trained on *sentences* instead of documents. For each entity-level pair, we merge the predictions of the mention-level pairs in different sentences using a maximum assumption: if at least one mention-level prediction indicates a positive relation between the target pair, then we predict the entity-concept pair as related, similarly to Gu et al. (2017). All of the described settings incorporate node type embeddings, contextual embeddings for MM edges and distance embeddings for MM and SS edges, unless otherwise stated.

We further compare our proposed approach to the state-of-the-art approaches on the CDR and the GDA datasets. For the CDR dataset, Verga et al. (2018), Gu et al. (2017) and Nguyen and Verspoor (2018) consider a single pair in each document,

[c] http://www.nactem.ac.uk/y-matsu/geniass/
[d] http://www.nactem.ac.uk/GENIA/tagger/

hence they repeat the input document a number of times that is equal to the number of entity-concept pairs in the document. Additionally, Verga et al. (2018) incorporates a Transformer-based network with sub-word embeddings and trains their proposed model on a section of the union of the training and development data. Gu et al. (2017) use two different networks for intra- and inter-sentence relations detection and merge the two outputs with additional rules in order to obtain the final overall performance. A similar case is that of Li et al. (2016b), who uses co-training with additional unlabelled training data. Peng et al. (2016) use an SVM-based method with hand-crafted features, additional training data and the shortest dependency path on the dependency graph of each sentence. Zhou et al. (2016a) proposed a combination of feature-based, kernel-based and neural network-based methods with additional hand-crafted features and post-processing rules. Panyam et al. (2018) proposed an SVM model with graph-based kernels, while Zheng et al. (2018) used a CNN stacked on top of an LSTM network with masked named entities. Finally, Sahu et al. (2019) proposed to apply a Graph Convolutional Neural network on the document-level dependency graph created following Quirk and Poon (2017).

It is important to mention that we consider a fair comparison with the model proposed by Sahu et al. (2019). For the remaining models, the training data and their pre-processing differ. For the GDA dataset, we draw comparisons with the model of Wu et al. (2019), which originally proposed the GDA dataset. In their method, they incorporate a CNN with a GRU-RNN network stacked on top. We re-trained their model on our own data split, for a fair comparison. However, they used entity masking, i.e. replaced the target named entities with a unique identifier (e.g. <CHEM> for chemicals and <DIS> for diseases) in their experiments. Their code was tailored to this choice and, as a result, the comparison between the two models is not completely fair.

6.4 Results

Table 6.3 depicts the performance of our proposed model on the CDR test set, in comparison with the state-of-the-art. The proposed model outperforms the (neural) state-of-the-art by 1.3 percentage points of overall performance. The models under the middle line correspond to approaches that take advantage of syntactic dependency tools or additional training data. Our model performs significantly better on intra- and inter-sentential pairs, even compared to models that rely on syntactic tools.

In comparison with the baselines, the EoG model performs best for all pair types.

Method	Overall (%)			Intra (%)			Inter (%)		
	P	R	F1	P	R	F1	P	R	F1
SVM+CNN (Gu et al., 2017)	55.7	68.1	61.3	59.7	55.0	57.2	51.9	7.0	11.7
Transformer (Verga et al., 2018)	55.6	70.8	62.1	-	-	-	-	-	-
CNN (Nguyen and Verspoor, 2018)	57.0	68.6	62.3	-	-	-	-	-	-
EoG	62.1	65.2	**63.6**	64.0	73.0	**68.2**	56.0	46.7	**50.9**
EoG (*Full*)	59.1	56.2	57.6$^{\diamond}$	71.2	62.3	66.5	37.1	42.0	39.4$^{\diamond}$
EoG (*NoInf*)	48.2	50.2	49.2$^{\diamond}$	65.8	55.2	60.2$^{\diamond}$	25.4	38.5	30.6$^{\diamond}$
EoG (*Sent*)	56.9	53.5	55.2$^{\diamond}$	56.9	76.4	65.2	-	-	-
Hybrid (Zhou et al., 2016a)	55.6	68.4	61.3	-	-	-	-	-	-
SVM (Peng et al., 2016)	62.1	64.2	63.1	-	-	-	-	-	-
SVM (Li et al., 2016b)	60.8	76.4	67.7	67.3	52.4	58.9	-	-	-
SVM (Panyam et al., 2018)	53.2	69.7	60.3	54.7	80.6	65.1	47.8	43.8	45.7
LSTM+CNN (Zheng et al., 2018)	56.2	67.9	61.5	-	-	-	-	-	-
GCN (Sahu et al., 2019)	52.8	66.0	58.6	-	-	-	-	-	-

Table 6.3: Overall, intra- and inter-sentence pairs performance comparison with the state-of-the-art on the CDR test set. The methods below the double line take advantage of additional training data and/or incorporate external tools. $\diamond$ indicates significance at $p < 0.01$ of the baselines compared with EoG.

In particular, for the inter-sentence pairs, performance significantly drops with a fully connected graph (*Full*) or without inference (*NoInf*). The former indicates that our heuristics are sensible. It is also important to note that the intra-sentence pairs substantially benefit from the document-level information, as EoG surpasses the performance of training on single sentences (*Sent*) by 3%. Finally, the performance drop in intra-sentence pairs, as a result of the inference algorithm removal (*NoInf*), suggests that multiple entity associations exist in sentences as well (Christopoulou et al., 2018). Their interactions can be beneficial in cases of lack of other context information.

Model	Dev. F1 (%)			Test F1 (%)		
	Overall	Intra	Inter	Overall	Intra	Inter
Wu et al. (2019)	80.0*	83.8*	55.7	81.1	84.8	56.2*
EoG	78.6	83.0	46.9	80.1	84.7	45.6
EoG (*Full*)	77.8*	82.2*	54.2*	79.9	84.6	54.7*
EoG (*NoInf*)	71.6*	77.1*	45.3	73.7*	79.2*	47.0
EoG (*Sent*)	72.1*	78.1*	-	73.0*	78.8*	-

Table 6.4: Performance comparison on the GDA development and test sets. $*$ indicates significance at $p < 0.05$ in comparison with *EoG*.

We also apply our model on the distantly supervised GDA dataset, as shown in Table 6.4, for both the development and the test sets. Our model results for intra-sentence

pairs are consistent with the findings of the CDR dataset for both development and test sets. This indicates that document-level information is helpful for intra-sentence RE. However, performance differs for inter-sentence pairs. In particular, it appears that the fully connected graph (*Full*) works best among the other settings, while the no inference mechanism is actually better than EoG on the test set. We partially attribute this behaviour to the small number of inter-sentence pairs in the GDA dataset (only 15% compared to 30% in the CDR dataset) that results in inadequate learning patterns for EoG. Another reason might be that since inter-sentence pairs are few, some interactions might already be identified during training as intra-sentence. In fact, 42% of inter-sentence pairs in the test set, are intra-sentence in the training set. This is also observed for CDR for 13% of inter-sentence pairs. We also believe that the automatic nature of the GDA dataset cannot guarantee that inter-sentence relations are correct in comparison with the human-annotated CDR dataset. Nevertheless, further investigation is required to determine the causes of this discrepancy between the two datasets. We leave further investigation as part of future work.

In comparison with the model proposed by Wu et al. (2019), we observe lower performance on both the development and the test sets. However, the comparison with our model is not absolutely fair, since we do not consider entity masking. We could not re-run their model without this feature. In general, we observe only a 2% difference in the inter-sentence performance (for the *Full* model), while similar intra-sentence performance is noted on the test set. By performing significance testing, we observe that the results between Wu et al. (2019) and our model are not significantly different on the test set for intra-sentence pairs.

Since we propose a document-level graph structure, it is reasonable to compare the performance model when using another graph encoder instead of our walk-based encoder. For this reason, we employ a Graph Convolutional Network (GCN) with residual connections between its layers and shared parameters from the second layer. The GCN setting relies on the node representations, which include the type of the node. In order to classify the entity-concept pairs, we simply concatenate the representations of the two entity-concept nodes. We additionally compare with the GCN model proposed by Sahu et al. (2019), that uses a dependency graph with additional co-reference edges. As shown in Table 6.5, our proposed model outperforms the dependency graph GCN for all relation pairs by a large margin. This might be due to the lack of contextualised representation into the GCN model. On the other hand, the GCN encoder on our

Model	Dev F1 (%)		
	Overall	Intra	Inter
GCN (1 Layer)	59.11*	65.68*	42.58
GCN (2 Layers)	60.15*	66.08*	44.55
GCN (3 Layers)	59.28*	65.11*	44.34
GCN (4 Layers)	59.15*	64.93*	43.93
EoG	63.57	68.25	46.68
Sahu et al. (2019)	57.19	63.43	36.90

Table 6.5: Comparison of the Edge-oriented Graph (EoG) with Graph Convolutional Network (GCN) on the CDR development set. $*$ indicates significance at $p < 0.05$ in comparison with EoG.

proposed graph performs better than the dependency graph. EoG is better by approximately 2% in intra- and 2% in inter-sentence relations in comparison with the best GCN model. However, application of other GCN variants (e.g. R-GCN (Schlichtkrull et al., 2018)) might yield better performance using the same graph structure. The small differences, overall, between the two encoders, indicate that the proposed graph structure, although simple, can be effective for detection of inter-sentence relations.

6.5 Analysis and Discussion

For further analysis, we choose the CDR dataset as it is human annotated. We conduct a series of experiments to better understand the advantages and limitations of the proposed model. We primarily analyse the effect of specific graph edges and then investigate some qualitative analysis.

6.5.1 Exploring the Effect of Edges

We conduct ablation analysis on the effect of direct and indirect sentence-to-sentence (SS) edges as a function of the walks length. Figures 6.3a, 6.3b and 6.3c illustrate the performance of both graphs for overall, intra- and inter-sentence pairs, respectively.

The first observation is that usage of direct SS edges only, reduces the overall performance almost by 4% for walks of length up-to $L = 8$. This drop mostly affects inter-sentence pairs, where an 18% point drop is observed. In fact, adjacent edges (SS_{direct}) need longer inference to perform better, in comparison with additional indirect edges (SS) for which less steps are required. The superiority of SS edges, for all

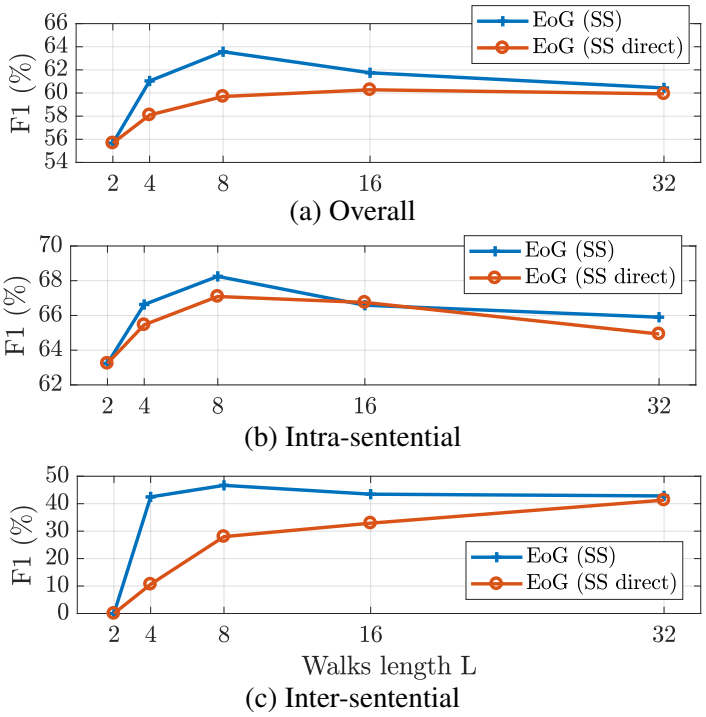


Figure 6.3: Performance as a function of the walks length when using direct (SS_{direct}) or direct and indirect (SS) sentence-to-sentence edges, on the CDR development set.

inference steps, compared to SS_{direct} edges on inter-sentence pairs detection, confirms our hypothesis that, in a narrative, some intermediate information might not be important or relevant. The observation that indirect edges perform slightly better than direct for intra-sentence pairs ($L \leq 16$) agrees with the results of Table 6.3, where we showed that inter-sentence information can act as complementary evidence for intra-sentence pairs.

We additionally conduct ablation analysis on the graph edges and nodes, as shown in Table 6.6. Usage of EE edges only (i.e. simple concatenation of Entity (E) node representations) results in poor performance across pairs, especially for intra-sentence ones. Information about intra-sentence pairs is implicitly encoded into the node representations via the mention nodes. However, the actual interactions between mentions in sentences are not explicitly encoded in this setting.

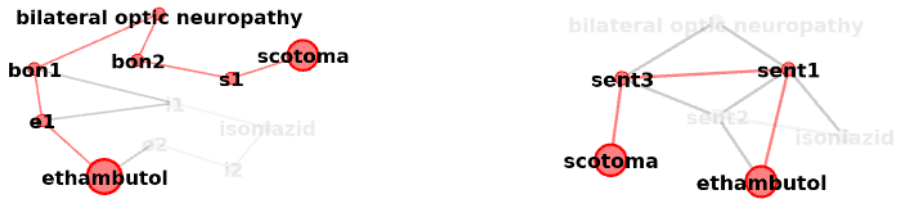
Removal of MM and ME edges does not significantly affect performance, as ES edges can replace their impact in the construction of EE edges. For instance, E-M-M-E is replaced by E-S-E. Complete removal of connections to M nodes results in low inter-sentence performance. This behaviour pinpoints that mention-level information and their interactions are important for the identification of cross-sentence relations.

Edge Types	F1 (%)		
	Overall	Intra	Inter
EE	55.14$^{\diamond}$	61.31$^{\diamond}$	40.34*
EoG	63.57	68.25	46.68
– MM	62.77	67.93	46.65
– ME	61.57$^{\diamond}$	66.39*	45.40
– MS	62.92	67.55	44.74
– ES	61.41*	66.44*	43.04
– $SS_{indirect}$	59.70$^{\diamond}$	67.09	28.00$^{\diamond}$
– SS	57.41$^{\diamond}$	65.45*	1.59$^{\diamond}$
– MM, ME, MS (M nodes)	60.46$^{\diamond}$	66.07*	39.56$^{\diamond}$
– ES, MS, SS (S nodes)	56.86$^{\diamond}$	64.63$^{\diamond}$	0.00

Table 6.6: Ablation analysis for different edge and node types on the CDR development set. $*$ and $\diamond$ indicate significance at $p < 0.05$ and $p < 0.01$ respectively, in comparison with EoG.

Removal of ES edges reduces the performance of all pairs, as encoding of EE edges becomes more challenging through longer walks. In practice, in this setting, we need paths of length 3 (E-M-M-E) for intra- and paths length 5 (E-M-S-S-M-E) for inter-sentence pairs. We further observe very poor identification of inter-sentence pairs without SS connections, either direct or indirect. This is complements the inability of the model to identify any inter-sentence pairs without connections to S nodes. In this scenario, we enable identification of pairs across sentences only through MM and ME edges, as shown in Figure 6.4a. In fact, for the CDR dataset, 78% of inter-sentential pairs have at least one argument that is mentioned only once in the document. The identification of these pairs, without S nodes, requires very long inference paths (minimum inference length 6, E-M-M-E-M-M-E). As shown in Figure 6.4b, the introduction of S nodes results in a path with half the length, which we expect to better represent the relation. Longer walk representations are weaker than shorter ones, as we have proved in previous chapters. We believe that this is also the case for the extraction of relations in documents. This suggests a limitation in the encoding mechanism and the graph structure, to model relations between far apart entities when there are no shorter (and valid) paths between them, e.g. in full-text documents.

The final analysis on the effect of edges is to investigate the additional enhancements incorporated in them, as shown in Table 6.7. In general, intra-sentence pairs are not affected by these settings. However, for inter-sentence pairs, removal of node type embeddings and distance embeddings results in a 2% and 5% drop in terms of



(a) MM, ME edges (b) ES, SS edges

Figure 6.4: Relation paths with different types of edges.

F1-score. These results indicate that the interactions between different elements in a document, along with the distance between sentences and mentions, play an important role in inter-sentence pair inference. Removing all of these settings does not perform worse than removing one of them, which might suggest potential model over-fitting due to the number of learned parameters.

Model	F1 (%)		
	Overall	Intra	Inter
EoG	63.57	68.25	46.68
– node types (T)	62.31	67.50	44.80
– MM context (C)	62.88	67.67	46.59
– distances (D)	62.53	68.00	41.53
– T, C, D	63.10	68.44	43.48

Table 6.7: Ablation analysis of edge enhancements on the CDR development set.

6.5.2 Supplementary Analysis

Moving on to other types of analysis, we further examine the performance of different models on detecting inter-sentence pairs, based on their distance in terms of intermediate sentences. Figure 6.5 illustrates that, as the distance increases, the performance of EoG decreases. In particular, for long-distanced pairs (distance ≥ 4), EoG has lower performance compared to the setting with a fully-connected graph. This points towards the difficulty in predicting far away pairs alongside a possible requirement for other, latent document-level information that EoG (*Full*) is able to capture.

We additionally report the learning curves of EoG and the proposed baselines in Figure 6.6. As we can observe, for almost all data samples, the proposed model outperforms all baselines. The first that seems to start saturating is the *Sent* model, probably

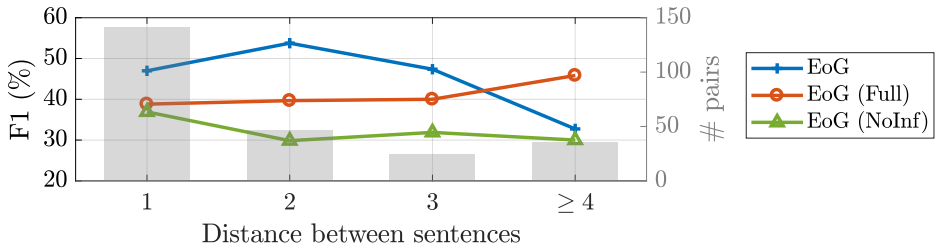


Figure 6.5: Performance of inter-sentence pairs on the CDR development set as a function of their sentence distance.

because it cannot take advantage of information from the entire document. On the contrary, *Full* and EoG have a very steep ascending trend, indicating that more patterns can be captured by the model when given more instances.

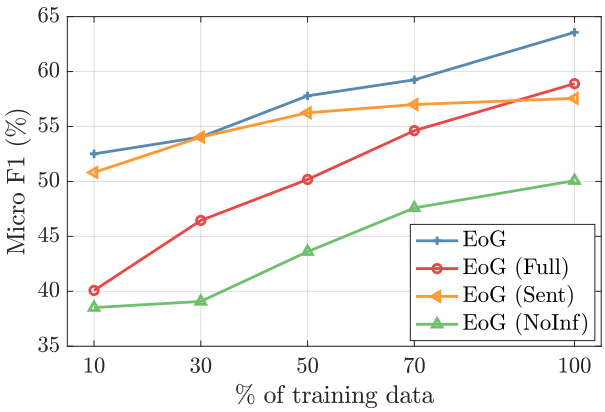


Figure 6.6: Learning curves for the proposed model and baselines on the CDR development set. The x-axis determines the percentage of training instances, where each instances is a document, expect for the case of *Sent* where it is a sentence.

Finally, we perform qualitative analysis and investigate some of the cases where the graph models succeed or fail to identify the related pairs. For this purpose, we randomly check some of the common false negative errors among the EoG models. We identify four types of errors, as shown in Table 6.8. In the first case, when multiple entities reside in the same sentence and are connected with conjunctions (e.g., 'and') or commas, the model often could not find associations with all of them. This can be likely because too many associations introduce noise. The second error derives from missing co-reference connections. For instance, *pyeloureteritis cystica* is referred to as *disease* in a later sentence. Although our model cannot directly create co-reference edges, S nodes potentially simulate such links by encoding the co-referring entities into the sentence representation. However, it is possible to augment the existing graph

Error	Snippet
Collocation	Following short exposure to oral **prednisone** [...]. Both presented in the emergency room with profound **coma**, **hypotension**, severe **hyperglycemia**, and acidosis.
Co-reference	The etiology of **pyeloureteritis cystica** has long been [...] The *disease* occurred subsequent to the initiation of **heparin** therapy [...]
Hypernymy	Time trends in **warfarin**-associated ***hemorrhage***. [...] The proportion of patients with major and ***intracranial bleeding*** increased [...]
Speculation	We suggest that **sleep deprivation** *may* add to the risk of **bupropion** - associated seizures .

Table 6.8: Examples of errors made by the EoG model.

if such information is available (e.g. a co-reference detection system is firstly applied on the document). The next error is associated with hyponymy detection. In the third example, *hemorrhage* is a hypernym of *intracranial bleeding* and, due to the MeSH hierarchy, they are assigned different KB IDs, hence treated as different entities. The model can find the intra-sentential relation but cannot generalise the inter-sentential one, as it has no knowledge of the hierarchy of terms. A potential solution is to add external ontology-related information into the model, in order to resolve such cases. The final example contains a speculative sentence, where the bold entities do not share a relation. However, our proposed model predicts them as related, ignoring the speculative term *may*.

6.6 Related Work

Traditional approaches focus on intra-sentence supervised RE, utilising CNN or RNN, ignoring multiple entities in a sentence (Zeng et al., 2014; Nguyen and Grishman, 2015) as well as incorporating external syntactic tools (Miwa and Bansal, 2016; Zhang et al., 2018b).

Other approaches deal with distantly-supervised datasets, but are also limited to intra-sentential relations. They utilise Piecewise Convolutional Neural Networks (PCNN) (Zeng et al., 2015), attention mechanisms (Lin et al., 2016; Zhou et al., 2018), entity descriptors (Jiang et al., 2016) and graph CNNs (Vashishth et al., 2018) to perform MIL on bags-of-sentences that contain multiple mentions of an entity pair. Recently,

Zeng et al. (2017b) proposed a method for extracting paths between entities, using the target entities' mentions in several different sentences (in possibly different documents), as intermediate connectors. They allow mention-to-mention edges only if these mentions belong to the same entity, considering that a single mention pair exists in a sentence and utilises only 1-hop inference. On the contrary, we not only allow interactions between all mentions in the same sentence, but also consider multiple edges between mentions, entities and sentences in a document with multi-hop inference.

Current approaches that try to deal with document-level RE are mostly graph-based. As discussed in Chapter 2, the work of (Quirk and Poon, 2017) introduced the notion of a document graph, where nodes are words and edges represent intra- and inter-sentential relations between the words. Following this work, other approaches incorporated graph-based models for document-level RE, such as graph LSTM (Peng et al., 2017), graph CNN (Song et al., 2018) or RNNs on dependency tree structures (Gupta et al., 2019). Recently, Jia et al. (2019) improved n-ary RE using information from multiple sentences and paragraphs in a document. Similar to our approach, they choose to directly classify concept-level pairs rather than multiple mention-level pairs. Although they consider sub-relations to model related tuples, they ignore interactions with other entities outside of the target tuple in the discourse units (sentence or paragraph).

Non-graph-based approaches utilise different intra- and inter-sentence models and merge the resulted predictions (Gu et al., 2016, 2017). Other approaches extract document-level representations for each candidate entity pair (Zheng et al., 2018; Li et al., 2018b; Wu et al., 2019), or use syntactic dependency structures (Zhou et al., 2016a; Peng et al., 2016). Verga et al. (2018) proposed a Transformer-based model for document-level relation extraction with multi-instance learning, merging multiple mention pairs. Nguyen and Verspoor (2018) used a CNN with additional character-level embeddings. Singh and Bhatia (2019) also utilised Transformer and connected two target entities by combining them directly and via a contextual token. However, they consider only a single target entity pair per document.

6.7 Conclusion

In this chapter, we presented a simple mechanism to transform documents into graphs and perform intra- and inter-sentence relation extraction simultaneously. The proposed approach constructs a partially connected, document-level graph with heterogeneous

types of nodes and edges. Our edge-oriented walk mechanism was applied on the graph structure with internal multi-instance learning. To the best of our knowledge, this is the first approach to utilise an edge-oriented model for document-level RE and construct graphs without dependencies for this task.

We experimented with two biomedical corpora, for Chemical-Disease and Gene-Disease interactions, and proved the effectiveness of the proposed graph structure without requiring external syntactic tools. The application of a GCN encoder instead of the walk-based inference, showed decent performance for the proposed graph structure, indicating that possibly more sophisticated graph encoding mechanisms can achieve better results.

Extensive analysis on the effect of different edges included in the graph, revealed that indirect connections between sentence nodes proved to be extremely useful for inter-sentence relation extraction. In addition, the introduction of sentence-level nodes allowed the generation of shorter connections between distant concepts. For both our tested datasets, we proved that document-level information can contribute to the identification of intra-sentence pairs leading to higher precision and F1-score.

We additionally discussed limitations of the current approach for a set specific linguistic phenomena such as speculation, co-reference and hypernymy. The explicit incorporation of external knowledge to the model, not only in the form of pre-trained word embeddings, can be effective for resolving most of these errors. Finally, another considered limitation is that, in the proposed approach, we assume named entity mentions and their associated concepts are given. Experiments on an automatically created corpus, however, with noisy annotations of mentions and entities, showed that the model can still produce reasonable performance which indicates that additional automatically generated data can be used to further train the model and provide domain-specific knowledge.

Chapter 7

Conclusions

This dissertation discussed the task of Relation Extraction, i.e. the identification of typed relations between named entities in textual sources. The main content was organised into seven chapters: An introductory chapter, an RE overview chapter, a technical background chapter, three main content chapters and the current (conclusion) chapter.

In the first part of the overview Chapter 2, we provided definitions of elements involved in Relation Extraction, discussed the multiple tasks that are currently studied, along with developed datasets, and described the evaluation metrics used. We defined four general categories under which we can group different relation extraction tasks, that correspond to the target domain, the number of arguments and their type, as well as the semantic categories of relations. In this dissertation we considered RE tasks from all of these categories. In particular, we delved into both the generic and the biomedical domain, identified relations in both sentences and paragraphs, while the entity types were named entity mentions or named entity concepts. Yet, our methods were restricted to the identification of pre-defined relation categories from existing datasets. The second part of the chapter explored the existing literature, categorising methods initially based on their type of learning and further based on the type of text structural representation they use. The latter was our main category of focus, where the effectiveness of recent graph-oriented approaches motivated our proposed RE methods.

Chapter 3 served mostly as a technical background chapter in an effort to familiarise non-expert readers with fundamental terminologies and techniques used in neural architectures. Since the main methodology of this dissertation is based on these tools, we briefly described the two basic architectures of RNNs and CNNs, attention mechanisms, as well as network parameter training techniques.

7.1 Confirmation of Research Hypotheses

The main goal of the presented work is to explore a different type of approaches for relation extraction, with a more specific focus to textual snippets that contain many annotated named entities. Our primary goal was to effectively map a textual snippet from a sequence into a graph structure using the contained named entities, and encode interactions between them by taking advantage of the graph form. As stated in the introduction, our approach is edge-oriented in the sense that it constructs multi-dimensional edge representations which are directly used to model interactions in a graph. The developed algorithm is able to encode a finite number of walks between two named entities, given a textual snippet, into the representation of an edge. This enables the proposed technique to represent associations between long-distance entities, as well as to identify interactions that are not necessarily supported by explicit context.

In Chapter 4, we addressed our initial research question and hypothesis regarding the modelling of multi-pair sentences as follows:

RQ_1 In cases where multiple entities exist in a sentence, could we take advantage of all entity-to-entity interactions to improve detection of semantic relations?

H_1 The relation between two named entities in a sentence can be supported by the interactions of these entities with other, co-existing named entities in the same sentence, in a joint training setting.

In order to test our hypothesis, we firstly investigated relation extraction between named entities in the generic domain, involving text from newswire articles and encyclopedia entries. Using all annotated entities in a sentence, we mapped the input into a fully connected graph structure, where nodes correspond to entities and edges correspond to the representations of relations between them. We introduced an edge-oriented graph model that forms interactions between pairs of entities as vectors. The core of the model is a walk-based mechanism that iteratively combines consecutive edges via an intermediate node, and forms edge representations that consider chains of entity interactions. The proposed mechanism is able to generate a new representation which essentially corresponds to a number of walks, of different lengths, between two nodes (entities) in the graph.

Evaluation on the proposed approach on three general domain datasets revealed that even if the method does not incorporate external syntactic parsers to assist RE, it

shows competitive results. By analysing the method, we observed that it was particularly successful when extracting relations from sentences that contained many entities, something that was consistent across datasets, thus confirming our hypothesis. In addition, we found that the method can outperform other methods that consider multiple entities per sentence when the entities are too far apart, showing that the edge-oriented walk algorithm can bridge the gap between long-distance pairs. We further observed that the proposed technique worked well, not only for multi-pair sentences, but also for sentences with only one pair. Performance consistently improved across datasets for such cases, proving that joint training of multiple interactions in a sentence can have a large positive impact for other pairs that do no have explicit additional information in their context.

Supplementary analysis of the walk-based mechanism revealed that longer walks are not so informative, hence they need to be weighted less into the edge representation, confirming existing findings about the shortest path being the most informative one in tree and graph structures. Ablation analysis on the information used to construct the initial edges showed that semantic entity types provide a strong inductive bias for relations. Furthermore, pair-centric context information into the edge representation appeared to not always be necessary. This suggested either a faulty choice of the context encoding mechanism or the redundancy of additional information.

Taking into account the aforementioned findings, we conclude the following:

- Detection of relations between entities can be supported by interactions with other entities in the same sentence.
- Joint training of multiple pair interactions in sentences, results in improved representations (and, consequently, performance) in both multi-pair and single-pair sentences.
- Shortest walks in graph structures contain more valuable information for relation extraction than longer ones.

In Chapter 5, we addressed the portability of the proposed graph-oriented approach to another domain, addressing our second research question:

RQ_2 Can multiple interactions among entities in a sentence be beneficial for detecting relations in other domains?

H_2 Modelling multiple interactions among pairs in a sentence can be effective for relations in both the generic and the biomedical domains.

To prove our hypothesis, we adapted the previously described edge-oriented model to the biomedical domain. In particular, we focused on two types of biomedical text: scientific articles and electronic health records. The first one studied interactions between Chemical and Gene/Protein entities, while the second one involved interactions between Drugs and other medication-related entities with a focus on ADEs. In this chapter, we first addressed the problem of representing pair-centric context information into the edge representation, where our previous mechanism proved to be faulty. We found that this information is beneficial for the biomedical domain, though using another attention mechanism to encode it. One of the findings was that there is a tendency that longer walks can benefit more from such information into the edge representations.

The proposed model also performed significantly better compared to a Transformer-based model, on both multi-pair and single-pair sentences for Chemical-Protein relations. Moreover, we found that adding certain types of interactions (e.g. Protein-Protein) in the graph had a positive impact on the performance. However, a limitation of our model was that it often failed to identify the existence of negation in a relation, especially in sentences where both positive and negative words occurred close to the target entities. This result revealed the need for more sophisticated context encoding mechanisms that can capture such phenomena.

In the case of EHRs, firstly, we again observed performance improvement in contrast with other models for multi- and single-pair sentences. Additionally, when we incorporated Drug-Drug interactions for the detection of Drug-medication relations, the detection of Reason-Drug and Frequency-Drug associations improved significantly. However, ADE-Drug associations still performed lower than other relations, due to their ambiguous nature in biomedical text. Inclusion of external resources as auxiliary information can assist the correct detection of such pairs. Overall, from all these experiments on sentence-level biomedical corpora, we confirm our hypothesis that interactions of named entities with other entities in a sentence can be beneficial across domains. We can thus conclude the following:

- Interactions between multiple pairs in sentences, can assist the detection of relations across domains.

The aforementioned work mainly focused on the context of sentences. As a result, in Chapter 6, we investigated the identification of relations across sentences, where it has already been proven that multiple interactions are helpful for the detection of inter-sentence relations (Quirk and Poon, 2017). We addressed our final research question

and hypotheses:

RH_3 Can we model documents as heterogeneous graph structures and infer document-level relations?

$H_{3.1}$ We can map documents to partially connected, heterogeneous graphs without the need for syntactic dependency structures.

$H_{3.2}$ Document-level inference, i.e. using information from the entire document, is beneficial for both intra- and inter-sentence relations.

We firstly proposed to create a different graph-structure from the sentence-level one, more suitable for documents. In particular, the graph was partially connected and included heterogeneous types of nodes and edges. Heuristics were used to connect different nodes in the graph and the proposed edge-oriented inference mechanism was applied. Another important difference is that we performed relation extraction between named entity concepts and not mentions, thus tackling the issue of entity surface form variation in biomedical text using multi-instance learning.

Evaluation on two document-level biomedical corpora revealed that our proposed approach was able to outperform existing approaches that were not graph-based, or incorporated other types of graphs (e.g. constructed from dependency parsers). We found that usage of a GCN encoder also provided decent performance with the proposed graph structure, thus confirming our first hypothesis. It is important to note that another, potentially more advanced, graph encoding mechanism can provide higher performance.

Analysis on the types of nodes and edges revealed that the most impactful nodes were the ones corresponding to sentences. In fact, the complete removal of sentence nodes disabled the detection of inter-sentence relations. We explained this phenomenon through the creation of "shortcuts" between named entities through sentence nodes. Their incorporation produced more robust edge representations, compared to only using entity and mention nodes, which resulted in the creation of very long inference paths. Consequently, we again confirmed our observation from sentences (Chapter 4), that longer walks are less informative compared to shorter ones. Furthermore, the introduction of indirect, non-adjacent, sentence-to-sentence edges was decisive for detection of inter-sentence associations. This finding supported a smaller hypothesis for the construction of these connections; that is, that intermediate sentences might

not be relevant or informative for the extraction of relations. Our second hypothesis was confirmed by observing that, for both corpora, the introduction of document level information compared to only using information from sentences yielded higher performance for intra-sentence relations.

Cases of errors produced by the model include missing co-reference links, identification of relations between hypernyms and not hyponyms, as well as wrongly classifying as positive speculative relations. Based on the above observations, we can draw the following conclusions:

- Document-level associations are important for both intra- and inter-sentence relations.
- We can construct document-level graphs without the need for external dependency tools and achieve good performance in detection of inter-sentence relations.

Overall, the work presented in this PhD thesis proved first, that graph structures are an effective structure representation for tasks that model connections between elements, such as relation extraction. Secondly, additional information from different interactions between pairs are beneficial for sentences and documents that contain two or more entities. We attempted to look at the task in the form of a graph, modelling edges as dense feature vectors. The success of the method comes with certain limitations, though we believe it might highlight a new angle on how pairs and their interactions can be represented. In addition, the proposed walk-based mechanism can further be used independently, on top of different architectures, in order to encode interactions between other types of nodes that are not necessarily named entities. We believe that the presented work will inspire further investigation of graph-based models and particularly how we can use other interactions in a present context to our advantage.

7.2 Limitations and Future Work

In this dissertation, we investigated only partially the task of Relation Extraction within and across sentences. Our proposed approach, although proved useful for the tasks at hand, has several limitations which should be addressed in future work.

7.2.1 Extension to Other Tasks

The proposed approach does not explicitly rely on any domain-specific resources and tools, such as POS taggers, dependency parsers or dictionaries. However, a major limitation of the work could consider that, in all of the methods, we assumed named entities and their semantic categories as given, regarding the case of sentence-level RE. For document-level RE, we also assumed that entity linking is performed in advance and mentions are grounded to KB IDs. Overall, we deemed that this choice is sensible given the recent advances in named entity recognition. However, it is expected that these assumptions will drastically affect the detection performance in more realistic scenarios. For this reason, we also performed experiments on noisy corpora, i.e. when annotations were produced automatically and not provided by experts, for both the case of sentences and documents. Despite a few discrepancies that are expected when the input is noisy, the proposed model performed reasonably. We believe that this is a strong indication that our proposed approach can work well without necessarily using gold annotations.

A certain step towards future work is to combine named entity recognition and relation extraction in an end-to-end manner, similarly to prior work (Roth and Yih, 2004; Miwa and Bansal, 2016; Bekoulis et al., 2018b). Standalone NER techniques already reach state-of-the-art performance. The joint training of named entities is generally beneficial for relation extraction. We expect that an end-to-end approach can benefit from our method, since it considers interactions between all entities in the sentence. This method is likely to force prediction of more entities, while simultaneously restrict relations between valid pairs of entities.

The method can also be extended to other tasks that can be converted into graphs. For instance, dependency parsing can be considered a relevant task. However, the number of nodes and consequently edges will be much larger, since the nodes correspond to the words in a sentence. A straightforward application of the document-level model can be for documents that do not have concept-level annotations. This can be realised by simply ignoring the concept (E) nodes. Additionally, at the current state, the model can be used to predict ternary relations, i.e. relations that consist of three arguments. This is realised in the walk-generation step that produces representations among three entities, which can be directly classified, before aggregation. Finally, we would like to adapt a semi-supervised direction in the future, in order to produce less data hungry models that can show better generalisation capabilities.

7.2.2 Memory Requirements

Another, more technical, drawback of the model is the amount of memory it requires to store the edge representations, mostly for document-level RE. If a graph contains N nodes, the overall number of edges is N^2. Since we consider nodes as entities (or other elements, but not words), the total number of nodes in our model is smaller than other approaches. However, we construct more representations. This can be restrictive in terms of memory usage, in cases were the graphs are constructed from full documents or the method is applied to Knowledge Base graphs.

A solution to this problem is to enable shared edge representations instead of using completely different representations for all of the edges in the graph. An interesting direction is to share some of the edge representations based on criteria that can either stem from heuristics, or from the graph structure itself. For example, we can share representations of edges that are similar to other edges using some similarity function between their vectors, or share edges between nodes that have a number of paths in common. Edge sharing might also result in improved performance since, despite our intuition that pairs depend on different contexts, there are several similarities between them in a textual snippet, which can be modelled via partial edge representation sharing.

7.2.3 Edge Engineering

In Chapters 4 and 5 we investigated different ways to construct an initial edge representation and, in more detail, how to incorporate additional context information into the representation. We found that the inclusion or exclusion of context has an impact on the model performance, although it is overall helpful mostly when using longer walks. An explanation might be that, when creating representations of long walks, the initial information of the edge is gradually forgotten and additional context can be proven useful.

In order to further investigate the impact of the edge representation one can perform *edge-engineering*. We investigated three types of attention mechanisms: vector (Zhou et al., 2016b), scale-dot (Vaswani et al., 2017) and argument-based (Wang et al., 2016) for context construction. However, we deem that the best choice of how one should encode the context of a pair (or incorporate additional context at all) largely depends on the task, the data at hand and the walk construction. Looking at recent approaches, bidirectional multi-head attention mechanisms (Devlin et al., 2019) can encode several

useful information from the context and, as a result, might be a more suitable choice for our method.

This is also related to edges constructed for document-level interactions, as described in Chapter 6. In these cases, since the graph is not fully connected, several properties of the graph can be used as additional edge-features. For instance, the degree of each node or the number of mentions residing in a sentence in mention-to-sentence edges. Instead of multi-dimensional features, we could convert this into edge weights which could be incorporated into the walk-based mechanism.

7.2.4 Walk-based Inference

Another limitation of our model, as pointed out in Chapter 4, is that the same walks length is used across all pairs in the sentence. However, different pairs typically require different inference, as indicated by our analysis on both the general and the biomedical domains. In order to address this, one should allow a different number of walks for each pair, which subsequently points to a potential pair-wise β weight. From our overall experiments, it appears that a value of β from 0.7 to 0.9 is effective for walks length larger or equal to 4, while walks of length 2 work well with a value of 0.5. However, this can be considered a hyper-parameter and, as such, be separately tuned or learned automatically during training. We believe that the latter is more suitable, since both the length of the walks and their weighting factor can be automatically adjusted for each dataset.

Another aspect of the walk-based mechanism is the aggregation step. Currently, we consider summing the different representations that result from usage of intermediate nodes. However, we could use a weighting scheme that can be learned during training to give different amount of importance to different walks. A simple scheme can again be an attention mechanism, that will weight the importance of each walk representation before aggregation.

On the other hand, we would like to investigate a combination of the edge-oriented approached with Graph Convolutional Neural models, due to their success in several tasks. So far, there have been several recent methods that take advantage of edge-based features in the form of weights (Beck et al., 2018; Vashishth et al., 2018; Schlichtkrull et al., 2018), thus an extension to multi-dimensional edge vectors seems interesting.

Appendix A

Hyper-parameter Settings

We report the hyper-parameter settings for each model in the aforementioned chapters with appropriate references.

A.1 Chapter 4

For the ACE 2005 dataset, we tuned the model hyper-parameters using the RoBO toolkit (Klein et al., 2017). The same hyper-parameters were used for the ACE 2004 dataset.

Parameter	
Optimisation Method	Bohamian
Maximiser	scipy
No iterations	30
Acquisition Function	log_ei
Acquisition Optimiser	L-BFGS-B
n_init	3
Learning rate	[0.001, 0.003]
Gradient clipping	[5, 30]
Input Layer dropout	[0.0, 0.5]
Output Layer dropout	[0.0, 0.5]
Regularisation	$[10^{-7}, 10^{-4}]$
Entity type dimension	10, 15, 20, 25
Relative position dimension	10, 15, 20, 25
β	[0.5, 0.9]

Table A.1: Tuning settings and hyper-parameter range for the ACE 2005 dataset.

The best parameters for each model are summarised in Table A.2

Parameter	ACE 2005 / 2004			
	$L=1$	$L=2$	$L=4$	$L=8$
β	-	0.72	0.77	0.88
Batchsize	10	10	10	10
Word dimension	200	200	200	200
Embedddings	Wikipedia / Google			
Attention	Additive			
LSTM dimension	100	100	100	100
Entity Type dimension	25	20	20	20
Relative Position dimension	25	25	25	25
Input Layer dropout	0.13	0.26	0.11	0.49
Output Layer dropout	0.38	0.38	0.32	0.36
Learning Rate	0.0017	0.003	0.002	0.001
Optimiser	Adam	Adam	Adam	Adam
Regularisation	$6.1{\cdot}10^{-5}$	0.0001	$5.7{\cdot}10^{-5}$	$1.88{\cdot}10^{-5}$
Gradient Clipping	30	8.6	24.4	10.5
Patience	5	5	5	5
Early stopping metric	Micro F1-score			
Parameter Averaging	✓	✓	✓	✓

Table A.2: Hyper-parameter settings of the walk-based model that were used for the ACE 2005 and ACE 2004 datasets, for different number of walks L.

For the Wikidata dataset, we selected hyper-parameters based on the best performing settings of the ACE dataset.

Parameter	WikiData	SemEval-2010
β	0.75	-
Batchsize	64	10
Word dimension	50	200
Attention	Additive	Additive
Embedddings	Glove 6B	Wikipedia
LSTM dimension	100	100
Entity Type dimension	-	-
Relative Position dimension	25	25
Input Layer dropout	0.5	0.0
Output Layer dropout	0.3	0.5
Learning Rate	0.002	0.003
Optimiser	Adam	Adam
Regularisation	10^{-5}	10^{-4}
Gradient Clipping	10	5
Patience	5	5
Early stopping metric	Validation Loss	Macro F1-score
Parameter Averaging	✓	✓

Table A.3: Hyper-parameter settings of the walk-based model for WikiData and SemEval-2010 datasets.

A.2 Chapter 5

For the n2c2 and ChemProt datasets, we tuned the model parameters with the RoBO toolkit (Klein et al., 2017), with the following settings.

Parameter	n2c2	ChemProt
Optimisation Method	Bohaiann	BayesOpt
Maximiser	scipy	scipy
No iterations	10	10
Acquisition Function	log_ei	ei
Acquisition Optimiser	L-BFGS-B	gp_mcmc
n_init	3	3
Learning rate	[0.001, 0.003]	[0.001, 0.003]
Gradient clipping	[5, 30]	[5, 30]
Input Layer dropout	[0.0, 0.5]	[0.0, 0.5]
Output Layer dropout	[0.0, 0.5]	[0.0, 0.5]
Regularisation	$[10^{-7}, 10^{-4}]$	$[10^{-7}, 10^{-4}]$
Entity type dimension	16, 20, 26, 32	10
Relative position dimension	15, 20, 25	25
β	[0.5, 0.9]	[0.5, 0.9]

Table A.4: Tuning settings and hyper-parameter range for the n2c2 and the ChemProt Datasets.

The final parameters used for both datasets are summarised in Table A.5.

A.3 Chapter 6

For the experiments conducted with the CDR and GDA datasets, we used the development set to identify the stopping training epoch and tune the number of inference iterations. Except from these parameters, all experiments used the same hyper-parameters, with a fixed initialisation seed. For the CDR dataset EoG, (*Full*) and (*Sent*) models performed best with $l = 8, 2, 4$ inference steps, respectively. The chosen batchsize was equal to 2. For the GDA dataset, EoG and EoG (*Sent*) performed best with $l = 16$ and EoG (*Full*) with $l = 4$ inference steps. The chosen batchsize was equal to 3. For all experiments performance was measured in terms of micro Precision (P), Recall (R) and F1-score (F1). We list the hyper-parameters used to train the proposed model in Table A.6.

Parameter	n2c2	ChemProt			
	$L=8$	$L=1$	$L=2$	$L=4$	$L=8$
β	0.75	-	0.5	0.76	0.85
Batchsize	3	10	10	10	10
Word dimension	200	200	200	200	200
Word Embeddings	Random	Random	Random	Random	Random
Attention	Scale-Dot	None	None	Scale-Dot	Scale-Dot
LSTM dimension	100	100	100	100	100
Entity Type dimension	26	10	10	10	10
Relative Position dimension	25	25	25	25	25
Input Layer dropout	0.46	0.5	0.0	0.008	0.33
Output Layer dropout	0.34	0.0	0.0	0.1	0.43
Learning Rate	0.002	0.003	0.003	0.003	0.0028
Optimiser	Adam	Adam	Adam	Adam	Adam
Regularisation	$2.6{\cdot}10^{-4}$	10^{-4}	10^{-4}	10^{-4}	$5.1{\cdot}10^{-5}$
Gradient Clipping	18.86	5	16.44	22.82	17.26
Patience	5	5	5	5	5
Early stopping metric	Micro F1	Micro F1	Micro F1	Micro F1	Micro F1
Parameter Averaging	✓	✓	✓	✓	✓

Table A.5: Hyper-parameter settings of the walk-based model for the n2c2 and the ChemProt BioCreative VI datasets for the corresponding number of walks L.

Parameter	CDR	GDA
β	0.8	0.8
Batch size	2	3
Word dimension	200	200
Embeddings	PubMed	Random
LSTM dimension	100	100
Edge dimension	100	100
Node type dimension	10	10
Inference iterations	$[0,5]$	$[0,5]$
Distance dimension	10	10
Dropout word embedding layer	0.5	0.5
Dropout classification layer	0.3	0.3
Learning rate	0.002	0.002
Optimiser	Adam	Adam
Regularisation	10^{-4}	10^{-4}
Gradient clipping	10	10
Early stop patience	10	5
Early stop metric	Micro F1	Micro F1
Parameter Averaging	✓	✓

Table A.6: Hyper-parameter settings used in the reported experiments for the CDR and the GDA datasets.

Bibliography

Heike Adel, Benjamin Roth, and Hinrich Schütze. Comparing convolutional neural networks to traditional models for slot filling. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 828–838. Association for Computational Linguistics, 2016.

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.

Antti Airola, Sampo Pyysalo, Jari Björne, Tapio Pahikkala, Filip Ginter, and Tapio Salakoski. A graph kernel for protein-protein interaction extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 1–9. Association for Computational Linguistics, 2008.

James F Allen and Alan M Frisch. What's in a semantic network? In *Proceedings of the 20th Annual Meeting on Association for Computational Linguistics*, pages 19–27. Association for Computational Linguistics, 1982.

Gabor Angeli, Julie Tibshirani, Jean Wu, and Christopher D. Manning. Combining distant and partial supervision for relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1556–1567. Association for Computational Linguistics, 2014.

Masaki Asada, Makoto Miwa, and Yutaka Sasaki. Extracting drug-drug interactions with attention CNNs. In *Proceedings of BioNLP*, pages 9–18. Association for Computational Linguistics, 2017.

Isabelle Augenstein, Diana Maynard, and Fabio Ciravegna. Relation extraction from the web using distant supervision. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 26–41. Springer, 2014.

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

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *Third International Conference on Learning Representations*, 2015.

Ivana Balazevic, Carl Allen, and Timothy Hospedales. TuckER: Tensor factorization for knowledge graph completion. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5185–5194. Association for Computational Linguistics.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. Matching the Blanks: Distributional Similarity for Relation Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905. Association for Computational Linguistics, 2019.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. Open Information Extraction from the Web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, pages 2670–2676. Morgan Kaufmann Publishers Inc., 2007.

David W Bates, David J Cullen, Nan Laird, Laura A Petersen, Stephen D Small, Deborah Servi, Glenn Laffel, Bobbie J Sweitzer, Brian F Shea, Robert Hallisey, et al. Incidence of adverse drug events and potential adverse drug events: implications for prevention. *JAMA*, 274(1):29–34, 1995.

Daniel Beck, Gholamreza Haffari, and Trevor Cohn. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–283. Association for Computational Linguistics, 2018.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. Adversarial training for multi-context joint entity and relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2830–2836. Association for Computational Linguistics, 2018a.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications*, 114:34 – 45, 2018b. ISSN 0957-4174.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

Nikita Bhutani, Yoshihiko Suhara, Wang-Chiew Tan, Alon Halevy, and H V Jagadish. Open information extraction from question-answer pairs. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2294–2305. Association for Computational Linguistics, 2019.

Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

Jari Björne and Tapio Salakoski. Biomedical Event Extraction Using Convolutional Neural Networks and Dependency Parsing. In *Proceedings of the BioNLP workshop*, pages 98–108. Association for Computational Linguistics, 2018.

Eduardo Blanco, Nuria Castell, and Dan Moldovan. Causal relation extraction. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC)*. European Language Resources Association (ELRA), 2008.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March 2003. ISSN 1532-4435.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 92–100, 1998.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795, 2013.

Karsten M Borgwardt and Hans-Peter Kriegel. Shortest-path kernels on graphs. In *Proceedings of the fifth IEEE International Conference on Data Mining*, pages 74–81. IEEE Computer Society, 2005.

Robert Bossy, Wiktoria Golik, Zorana Ratkovic, Philippe Bessières, and Claire Nédellec. BioNLP shared task 2013 – An Overview of the Bacteria Biotope task. In *Proceedings of the BioNLP Shared Task Workshop*, pages 161–169. Association for Computational Linguistics, 2013.

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

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.

Razvan C Bunescu and Raymond J Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 724–731. Association for Computational Linguistics, 2005.

Razvan C Bunescu and Raymond J Mooney. Subsequence kernels for relation extraction. In *Advances in Neural Information Processing Systems*, pages 171–178, 2006.

Rui Cai, Xiaodong Zhang, and Houfeng Wang. Bidirectional recurrent convolutional neural network for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 756–765. Association for Computational Linguistics, 2016.

Marc-André Carbonneau, Veronika Cheplygina, Eric Granger, and Ghyslain Gagnon. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 77:329–353, 2018.

Rich Caruana, Steve Lawrence, and C. Lee Giles. Overfitting in neural nets: Backpropagation, Conjugate Gradient, and Early stopping. In *Advances in Neural Information Processing Systems*, pages 402–408. MIT Press, 2001.

Augustin Cauchy. Méthode générale pour la résolution des systemes d'équations simultanées. *Comptes rendus de l'Académie des Sciences Paris*, 25(1847):536–538, 1847.

Matthias Cetto, Christina Niklaus, André Freitas, and Siegfried Handschuh. Graphene: Semantically-linked propositions in open information extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2300–2311. Association for Computational Linguistics, 2018.

Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. Unsupervised feature selection for relation extraction. In *Companion Volume to the Proceedings of Conference including Posters/Demos and Tutorial Abstracts*, 2005.

Billy Chiu, Gamal Crichton, Anna Korhonen, and Sampo Pyysalo. How to Train good Word Embeddings for Biomedical NLP. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 166–174. Association for Computational Linguistics, 2016.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734. Association for Computational Linguistics, 2014.

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: Short Papers)*, pages 81–88. Association for Computational Linguistics, 2018.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 4924–4935. Association for Computational Linguistics, 2019.

Fenia Christopoulou, Thy Thy Tran, Sunil Kumar Sahu, Makoto Miwa, and Sophia Ananiadou. Adverse drug events and medication relation extraction in electronic health records with ensemble deep learning methods. *Journal of the American Medical Informatics Association*, 27(1):39–46, 2020.

Jonathan H Clark and José P González-Brenes. Coreference resolution: Current trends and future directions. *Language and Statistics II Literature Review*, pages 1–14, 2008.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy, 2019. Association for Computational Linguistics.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167. ACM, 2008.

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.

Peter Corbett and John Boyle. Improving the learning of chemical-protein interactions from literature using transfer learning and specialized word embeddings. *Database*, 2018.

Mark Craven, Johan Kumlien, et al. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 77–86, 1999.

Nello Cristianini, John Shawe-Taylor, et al. *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

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: Short Papers)*, pages 407–413. Association for Computational Linguistics, 2018a.

Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering*, 31(5):833–852, 2018b.

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

Ido Dagan and Oren Glickman. Probabilistic textual entailment: Generic applied modeling of language variability. *Learning Methods for Text Understanding and Mining*, 2004:26–29, 2004.

Bharath Dandala, Diwakar Mahajan, and Murthy V Devarakonda. IBM research system at TAC 2017: Adverse Drug Reactions Extraction from Drug Labels. In *TAC*, 2017.

Bharath Dandala, Venkata Joopudi, and Murthy Devarakonda. Ibm research system at made 2018: Detecting adverse drug events from electronic health records. In *International Workshop on Medication and Adverse Drug Event Detection*, pages 39–47, 2018.

Rajarshi Das, Arvind Neelakantan, David Belanger, and Andrew McCallum. Chains of reasoning over entities, relations, and text using recurrent neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 132–141. Association for Computational Linguistics, 2017.

Tirthankar Dasgupta, Abir Naskar, and Lipika Dey. Exploring linguistic and graph based features for the automatic classification and extraction of adverse drug effects. In *International Conference on Computational Linguistics and Intelligent Text Processing*, pages 463–474. Springer, 2017.

Berry de Bruijn, Colin Cherry, Svetlana Kiritchenko, Joel Martin, and Xiaodan Zhu. Nrc at i2b2: one challenge, three practical tasks, nine statistical systems, hundreds of clinical records, millions of useful features. In *Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data. Boston, MA, USA: i2b2*, 2010.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Question answering by Reasoning Across Documents with Graph Convolutional Networks. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2306–2317. Association for Computational Linguistics, 2019.

Michael E DeBakey. The National Library of Medicine: evolution of a premier information center. *JAMA*, 266(9):1252–1258, 1991.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

Ika Novita Dewi, Shoubin Dong, and Jinlong Hu. Drug-drug interaction relation extraction with deep convolutional neural networks. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1795–1802. IEEE, 2017.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*. European Language Resources Association (ELRA), 2004.

Cícero 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: Long Papers)*, pages 626–634. Association for Computational Linguistics, 2015a.

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: Long Papers)*, pages 626–634. Association for Computational Linguistics, 2015b.

Jinhua Du, Jingguang Han, Andy Way, and Dadong Wan. Multi-level structured self-attentions for distantly supervised relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2216–2225. Association for Computational Linguistics, 2018.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12 (Jul):2121–2159, 2011.

Javid Ebrahimi and Dejing Dou. Chain based RNN for relation classification. In *Proceedings of the Conference of the North American Chapter of the Association*

for Computational Linguistics: Human Language Technologies, pages 1244–1249. Association for Computational Linguistics, 2015.

Jeffrey L Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*. European Language Resources Association (ELRA), 2018.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, et al. Open information extraction: The second generation. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics, 2011.

Hong Li Feiyu Xu, Hans Uszkoreit and Niko Felger. Adaptation of relation extraction rules to new domains. In Bente Maegaard Joseph Mariani Jan Odijk Stelios Piperidis Daniel Tapias Nicoletta Calzolari (Conference Chair), Khalid Choukri, editor, *Proceedings of the Sixth International Conference on Language Resources and Evaluation*. European Language Resources Association (ELRA), 2008.

Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. Reinforcement learning for relation classification from noisy data. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Lisheng Fu and Ralph Grishman. An efficient active learning framework for new relation types. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 692–698. Asian Federation of Natural Language Processing, 2013.

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: Short Papers)*, pages 425–429, 2017.

Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. GraphRel: Modeling Text as Relational Graphs for Joint Entity and Relation Extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1409–1418. Association for Computational Linguistics, 2019.

Ryan Gabbard, Marjorie Freedman, and Ralph Weischedel. Coreference for Learning to Extract Relations: Yes, Virginia, Coreference Matters. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 288–293. Association for Computational Linguistics, 2011.

Pablo Gamallo, Marcos Garcia, and Santiago Fernández-Lanza. Dependency-based open information extraction. In *Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP*, pages 10–18, Avignon, France, 2012. Association for Computational Linguistics.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. FewRel 2.0: Towards more challenging few-shot relation classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 6250–6255. Association for Computational Linguistics, 2019.

Anna Gaulton, Louisa J Bellis, A Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, et al. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, 40(D1):D1100–D1107, 2011.

Jeremy Getman, Joe Ellis, Stephanie Strassel, Zhiyi Song, and Jennifer Tracey. Laying the groundwork for knowledge base population: Nine years of linguistic resources for TAC KBP. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*. European Language Resources Association (ELRA), 2018.

Claudio Giuliano, Alberto Lavelli, and Lorenza Romano. Exploiting shallow linguistic information for relation extraction from biomedical literature. In *11th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2006.

Pavel Golik, Patrick Doetsch, and Hermann Ney. Cross-entropy vs. squared error training: a theoretical and experimental comparison. In *Interspeech*, volume 13, pages 1756–1760, 2013.

Liyu Gong and Qiang Cheng. Exploiting edge features for graph neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9211–9219, 2019.

Matthew R. Gormley, Mo Yu, and Mark Dredze. Improved relation extraction with feature-rich compositional embedding models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1774–1784. Association for Computational Linguistics, 2015.

Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6): 602–610, 2005.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864. ACM, 2016.

Jinghang Gu, Longhua Qian, and Guodong Zhou. Chemical-induced disease relation extraction with various linguistic features. *Database*, 2016.

Jinghang Gu, Fuqing Sun, Longhua Qian, and Guodong Zhou. Chemical-induced disease relation extraction via convolutional neural network. *Database*, 2017.

Zhijiang Guo, Yan Zhang, and Wei Lu. Attention Guided Graph Convolutional Networks for Relation Extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 241–251. Association for Computational Linguistics, 2019.

Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2537–2547. The COLING Organizing Committee, 2016.

Pankaj Gupta, Subburam Rajaram, Hinrich Schütze, and Thomas Runkler. Neural Relation Extraction within and across Sentence Boundaries. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6513–6520, 2019.

Harsha Gurulingappa, Abdul Mateen-Rajpu, and Luca Toldo. Extraction of potential adverse drug events from medical case reports. *Journal of Biomedical Semantics*, 3 (1):15, 2012a.

Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of Biomedical Informatics*, 45(5):885–892, 2012b.

Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R. Curran. Evaluating entity linking with wikipedia. *Artificial Intelligence*, 194:130 – 150, 2013. ISSN 0004-3702. Artificial Intelligence, Wikipedia and Semi-Structured Resources.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. FewRel: A Large-Scale Supervised Few-Shot Relation Classification Dataset with State-of-the-Art Evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809. Association for Computational Linguistics, 2018.

Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 415–422, 2004.

Kazuma Hashimoto, Makoto Miwa, Yoshimasa Tsuruoka, and Takashi Chikayama. Simple customization of recursive neural networks for semantic relation classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1372–1376. Association for Computational Linguistics, 2013.

Donald Olding Hebb. *The organization of behavior: a neuropsychological theory*. Science Editions, 1962.

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 5th International Workshop on Semantic Evaluation*, pages 33–38. Association for Computational Linguistics, 2010.

Sam Henry and Bridget T McInnes. Literature based discovery: models, methods, and trends. *Journal of Biomedical Informatics*, 74:20–32, 2017.

Sam Henry, Kevin Buchan, Michele Filannino, Amber Stubbs, and Ozlem Uzuner. 2018 n2c2 shared task on adverse drug events and medication extraction in electronic health records. *Journal of the American Medical Informatics Association*, 27 (1):3–12, 10 2019.

Karl Moritz Hermann and Phil Blunsom. The role of syntax in vector space models of compositional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 894–904. Association for Computational Linguistics, 2013.

María Herrero-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. The ddi corpus: An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of Biomedical Informatics*, 46(5):914–920, 2013.

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

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

FP Hogenboom, F Frasincar, U Kaymak, and FMG Jong, de. An overview of event extraction from text. pages 48–57, 2011.

David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1): 106–154, 1962.

Heikki Hyotyniemi. Turing machines are recurrent neural networks. 1996.

Ander Intxaurrondo, Mihai Surdeanu, Oier Lopez De Lacalle, and Eneko Agirre. Removing noisy mentions for distant supervision. *Procesamiento del Lenguaje Natural*, (51):41–48, 2013.

Abhyuday Jagannatha, Feifan Liu, Weisong Liu, and Hong Yu. Overview of the first natural language processing challenge for extracting medication, indication, and adverse drug events from electronic health record notes (MADE 1.0). *Drug Safety*, 42 (1):99–111, 2019.

Sarthak Jain and Byron C Wallace. Attention is not Explanation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556. Association for Computational Linguistics, 2019.

David M Jessop, Sam E Adams, Egon L Willighagen, Lezan Hawizy, and Peter Murray-Rust. OSCAR4: A flexible architecture for chemical text-mining. *Journal of Cheminformatics*, 3(1):41, 2011.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3060–3066. AAAI Press, 2017.

Robin Jia, Cliff Wong, and Hoifung Poon. Document-level n-ary relation extraction with multiscale representation learning. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3693–3704. Association for Computational Linguistics, 2019.

Jing Jiang. Multi-task transfer learning for weakly-supervised relation extraction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1012–1020. Association for Computational Linguistics, 2009.

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

Xiaotian Jiang, Quan Wang, Peng Li, and Bin Wang. Relation extraction with multi-instance multi-label convolutional neural networks. In *Proceedings of COLING,*

the 26th International Conference on Computational Linguistics: Technical Papers, pages 1471–1480. The COLING Organizing Committee, 2016.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3:160035, 2016.

Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *International Conference on Machine Learning*, pages 2342–2350, 2015.

Nanda Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the Association for Computational Linguistics on Interactive poster and demonstration sessions*, page 22. Association for Computational Linguistics, 2004.

Minoru Kanehisa and Susumu Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 28(1):27–30, 2000.

Ning Kang, Bharat Singh, Chinh Bui, Zubair Afzal, Erik M van Mulligen, and Jan A Kors. Knowledge-based extraction of adverse drug events from biomedical text. *BMC Bioinformatics*, 15(1):64, 2014.

Arzoo Katiyar and Claire Cardie. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 917–928. Association for Computational Linguistics, 2017.

Sophia Katrenko and Pieter Adriaans. Learning relations from biomedical corpora using dependency trees. In *International Workshop on Knowledge Discovery and Emergent Complexity in Bioinformatics*, pages 61–80. Springer, 2006.

Ramakanth Kavuluru, Anthony Rios, and Tung Tran. Extracting drug-drug interactions with word and character-level recurrent neural networks. In *IEEE International Conference on Healthcare Informatics (ICHI)*, pages 5–12, 2017.

Halil Kilicoglu and Sabine Bergler. Recognizing speculative language in biomedical research articles: a linguistically motivated perspective. *BMC Bioinformatics*, 9 (S11):S10, 2008.

Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751. Association for Computational Linguistics, 2014.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2015.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*. OpenReview.net, 2017.

A. Klein, S. Falkner, N. Mansur, and F. Hutter. Robo: A flexible and robust bayesian optimization framework in python. In *Proceedings of Workshop on Bayesian Optimization in the Conference on Neural Information Processing Systems*, 2017.

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

Martin Krallinger, Obdulia Rabal, Saber A Akhondi, et al. Overview of the BioCreative VI chemical-protein interaction track. In *Proceedings of the sixth BioCreative Challenge Evaluation Workshop*, volume 1, pages 141–146, 2017.

Robert Leaman, Rezarta Islamaj Doğan, and Zhiyong Lu. Dnorm: disease name normalization with pairwise learning to rank. *Bioinformatics*, 29(22):2909–2917, 2013.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

Fei Li, Meishan Zhang, Guohong Fu, and Donghong Ji. A neural joint model for entity and relation extraction from biomedical text. *BMC Bioinformatics*, 18(1):198, 2017.

Fei Li, Weisong Liu, and Hong Yu. Extraction of information related to Adverse Drug Events from Electronic Health Record notes: Design of an end-to-end model based on deep learning. *JMIR Medical Informatics*, 6(4):e12159, 2018a.

Haodi Li, Ming Yang, Qingcai Chen, Buzhou Tang, Xiaolong Wang, and Jun Yan. Chemical-induced disease extraction via recurrent piecewise convolutional neural networks. *BMC Medical Informatics and Decision Making*, 18(2):60, 2018b.

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database*, 2016a.

Jiwei Li, Thang Luong, Dan Jurafsky, and Eduard Hovy. When are tree structures necessary for deep learning of representations? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2304–2314. Association for Computational Linguistics, 2015.

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

Zhiheng Li, Zhihao Yang, Hongfei Lin, Jian Wang, Yingyi Gui, Yin Zhang, and Lei Wang. CIDextractor: A chemical-induced disease relation extraction system for biomedical literature. In *IEEE International Conference on Bioinformatics and Biomedicine*, pages 994–1001. IEEE, 2016b.

Sangrak Lim and Jaewoo Kang. Chemical–gene relation extraction using recursive neural network. *Database*, 2018, 2018.

Sangrak Lim, Kyubum Lee, and Jaewoo Kang. Drug drug interaction extraction from the literature using a recursive neural network. *PLOS ONE*, 13(1):1–17, 01 2018.

Dekang Lin and Patrick Pantel. DIRT@ SBT@ discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 323–328. ACM, 2001.

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

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. In *International Conference on Learning Representations, ICLR*. OpenReview.net, 2017.

Xiao Ling and Daniel S Weld. Temporal information extraction. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

Bing Liu, Longhua Qian, Hongling Wang, and Guodong Zhou. Dependency-driven feature-based learning for extracting protein-protein interactions from biomedical text. In *COLING 2010: Posters*, pages 757–765. COLING Organizing Committee, 2010.

Mei Liu, Michael E Matheny, Yong Hu, and Hua Xu. Data mining methodologies for pharmacovigilance. *ACM SIGKDD Explorations Newsletter*, 14(1):35–42, 2012.

Ruifeng Liu, Mohamed Diwan M AbdulHameed, Kamal Kumar, Xueping Yuˆ, Anders Wallqvist, and Jaques Reifman. Data-driven prediction of adverse drug reactions induced by drug-drug interactions. *BMC Pharmacology and Toxicology*, 18(1):44, 2017a.

Shengyu Liu, Buzhou Tang, Qingcai Chen, and Xiaolong Wang. Drug-drug interaction extraction via convolutional neural networks. *Computational and Mathematical Methods in Medicine*, 2016:8, 2016. ISSN 1748-670X.

Sijia Liu, Feichen Shen, Yanshan Wang, Majid Rastegar-Mojarad, Ravikumar Komandur Elayavilli, Vipin Chaudhary, and Hongfang Liu. Attention-based neural networks for chemical protein relation extraction. *Training*, 1020(25.247):4157, 2017b.

Sijia Liu, Feichen Shen, Ravikumar Komandur Elayavilli, Yanshan Wang, Majid Rastegar-Mojarad, Vipin Chaudhary, and Hongfang Liu. Extracting chemical–protein relations using attention-based neural networks. *Database*, 2018, 2018.

Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng WANG. A dependency-based neural network for relation classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 285–290. Association for Computational Linguistics, 2015.

Oier Lopez de Lacalle and Mirella Lapata. Unsupervised relation extraction with general domain knowledge. In *Proceedings of the Conference on Empirical Methods*

in Natural Language Processing, pages 415–425. Association for Computational Linguistics, 2013.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. A general framework for information extraction using dynamic span graphs. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3036–3046. Association for Computational Linguistics, 2019.

Pei-Yau Lung, Zhe He, Tingting Zhao, Disa Yu, and Jinfeng Zhang. Extracting chemical–protein interactions from literature using sentence structure analysis and feature engineering. *Database*, 2019.

Bingfeng Luo, Yansong Feng, Zheng Wang, Zhanxing Zhu, Songfang Huang, Rui Yan, and Dongyan Zhao. Learning with noise: Enhance distantly supervised relation extraction with dynamic transition matrix. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 430–439. Association for Computational Linguistics, 2017.

Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. Association for Computational Linguistics, September 2015.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60. Association for Computational Linguistics, 2014.

Diego Marcheggiani and Ivan Titov. Discrete-state variational autoencoders for joint discovery and factorization of relations. *Transactions of the Association for Computational Linguistics*, 4:231–244, 2016.

Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark, 2017. Association for Computational Linguistics.

Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero. A survey of link prediction in complex networks. *ACM Computing Surveys (CSUR)*, 49(4):69, 2017.

Carolyn J Mattingly, Michael C Rosenstein, Allan Peter Davis, Glenn T Colby, John N Forrest Jr, and James L Boyer. The comparative toxicogenomics database: a cross-species resource for building chemical-gene interaction networks. *Toxicological Sciences*, 92(2):587–595, 2006.

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

Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of Mathematical Biophysics*, 5(4):115–133, 1943.

Ryan McDonald, Fernando Pereira, Seth Kulick, Scott Winters, Yang Jin, and Pete White. Simple algorithms for complex relation extraction with applications to biomedical ie. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 491–498. Association for Computational Linguistics, 2005.

MedLine. https://www.nlm.nih.gov/bsd/pmresources.html.

Farrokh Mehryary, Jari Björne, Tapio Salakoski, and Filip Ginter. Combining support vector machines and LSTM networks for chemical protein relation extraction. In *Proceedings of the BioCreative VI Workshop*, pages 176–180, 2017.

Farrokh Mehryary, Jari Björne, Tapio Salakoski, and Filip Ginter. Potent pairing: ensemble of long short-term memory networks and support vector machine for chemical-protein relation extraction. *Database*, 2018.

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

George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

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

Marvin Minsky and Seymour Papert. An introduction to computational geometry. *Cambridge tiass., HIT*, 1969.

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 Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP (Volume 2)*, pages 1003–1011. Association for Computational Linguistics, 2009.

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: Long Papers)*, pages 1105–1116. Association for Computational Linguistics, 2016.

Makoto Miwa and Yutaka Sasaki. Modeling joint entity and relation extraction with table representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1858–1869. Association for Computational Linguistics, 2014.

Makoto Miwa, Rune Sætre, Yusuke Miyao, and Jun'ichi Tsujii. Protein–protein interaction extraction by leveraging multiple kernels and parsers. *International Journal of Medical Informatics*, 78(12):e39 – e46, 2009. ISSN 1386-5056. Mining of Clinical and Biomedical Text and Data Special Issue.

Yusuke Miyao, Rune Sætre, Kenji Sagae, Takuya Matsuzaki, and Jun'ichi Tsujii. Task-oriented evaluation of syntactic parsers and their representations. In *Proceedings of Association for Computational Linguistics: Human Language Technologies*, pages 46–54. Association for Computational Linguistics, 2008.

Thahir P Mohamed, Jaime G Carbonell, and Madhavi K Ganapathiraju. Active learning for human protein-protein interaction prediction. *BMC Bioinformatics*, 11(1): S57, 2010.

Andrés Montoyo, Patricio Martínez-Barco, and Alexandra Balahur. Subjectivity and sentiment analysis: An overview of the current state of the area and envisaged developments. *Decision Support Systems*, 53(4):675 – 679, 2012.

Tsendsuren Munkhdalai, Feifan Liu, and Hong Yu. Clinical relation extraction toward drug safety surveillance using Electronic Health Record narratives: classical learning versus deep learning. *JMIR Public Health and Surveillance*, 4(2):e29, 2018.

Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. Patty: a taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1135–1145. Association for Computational Linguistics, 2012.

Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. Compositional vector space models for knowledge base completion. 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: Long Papers)*, pages 156–166. Association for Computational Linguistics, 2015.

Dat PT Nguyen, Yutaka Matsuo, and Mitsuru Ishizuka. Relation extraction from wikipedia using subtree mining. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, page 1414. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.

Dat Quoc Nguyen. An overview of embedding models of entities and relationships for knowledge base completion. *arXiv preprint arXiv:1703.08098*, 2017.

Dat Quoc Nguyen and Karin Verspoor. Convolutional neural networks for chemical-disease relation extraction are improved with character-based word embeddings. In *Proceedings of the BioNLP workshop*, pages 129–136. Association for Computational Linguistics, 2018.

Minh Luan Nguyen, Ivor W Tsang, Kian Ming Adam Chai, and Hai Leong Chieu. Robust domain adaptation for relation extraction via clustering consistency. In *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014-Proceedings of the Conference*, 2014.

Thien Huu Nguyen and Ralph Grishman. Employing word representations and regularization for domain adaptation of relation extraction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 68–74, 2014.

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.

Thien Huu Nguyen, Barbara Plank, and Ralph Grishman. Semantic representations for domain adaptation: A case study on the tree kernel-based method for relation 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: Long Papers)*, pages 635–644, 2015.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. A survey on open information extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3866–3878. Association for Computational Linguistics, 2018.

Eric W Noreen. *Computer-intensive methods for testing hypotheses*. Wiley New York, 1989.

Yuriy Ostapov. Question answering in a natural language understanding system based on object-oriented semantics. *arXiv preprint arXiv:1111.4343*, 2011.

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

Nagesh C Panyam, Karin Verspoor, Trevor Cohn, and Kotagiri Ramamohanarao. Exploiting graph kernels for high performance biomedical relation extraction. *Journal of Biomedical Semantics*, 9(1):7, 2018.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1310–1318. PMLR, 2013.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

Sachin Pawar, Pushpak Bhattacharyya, and Girish Keshav Palshikar. Semi-supervised relation extraction using EM algorithm. In *International Conference on Natural Language Processing (ICON)*. NLP Association of India, 2014.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. Cross-sentence n-ary relation extraction with graph LSTMs. *Transactions of the Association for Computational Linguistics*, 5:101–115, 2017.

Yifan Peng, Chih-Hsuan Wei, and Zhiyong Lu. Improving chemical disease relation extraction with rich features and weakly labeled data. *Journal of Cheminformatics*, 8(1):53, 2016.

Yifan Peng, Anthony Rios, Ramakanth Kavuluru, and Zhiyong Lu. Chemical-protein relation extraction with ensembles of SVM, CNN, and RNN models. *Database: The Journal of Biological Databases and Curation*, 2018.

Yifan Peng, Shankai Yan, and Zhiyong Lu. Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 58–65. Association for Computational Linguistics, August 2019.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining*, pages 701–710. ACM, 2014.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics, 2018.

Janet Piñero, Àlex Bravo, Núria Queralt-Rosinach, Alba Gutiérrez-Sacristán, Jordi Deu-Pons, Emilio Centeno, Javier García-García, Ferran Sanz, and Laura I Furlong. DisGeNET: a comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic Acids Research*, pages D833–D839, 2016.

Barbara Plank and Alessandro Moschitti. Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1498–1507, 2013.

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

Sampo Pyysalo, Antti Airola, Juho Heimonen, Jari Björne, Filip Ginter, and Tapio Salakoski. Comparative analysis of five protein-protein interaction corpora. In *BMC Bioinformatics*, volume 9, page S6. BioMed Central, 2008.

Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145–151, 1999.

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: Long Papers)*, pages 496–505. Association for Computational Linguistics, 2018.

Chanqin Quan, Lei Hua, Xiao Sun, and Wenjun Bai. Multichannel convolutional neural network for biological relation extraction. *BioMed Research International*, 2016, 2016.

Chris Quirk and Hoifung Poon. Distant supervision for relation extraction beyond the sentence boundary. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1171–1182. Association for Computational Linguistics, 2017.

Desh Raj, Sunil Sahu, and Ashish Anand. Learning local and global contexts using a convolutional recurrent network model for relation classification in biomedical text. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL)*, pages 311–321. Association for Computational Linguistics, 2017.

Suriyadeepan Ramamoorthy and Selvakumar Murugan. An attentive sequence model for adverse drug event extraction from biomedical text. *arXiv preprint arXiv:1801.00625*, 2018.

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

Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In José Luis Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 148–163, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979. ISBN 0408709294.

Thomas C Rindflesch, Lorraine Tanabe, John N Weinstein, and Lawrence Hunter. EDGAR: extraction of drugs, genes and relations from the biomedical literature. In *Pacific Symposium on Biocomputing*, pages 517–528. World Scientific, 2000.

Anthony Rios, Ramakanth Kavuluru, and Zhiyong Lu. Generalizing biomedical relation classification with neural adversarial domain adaptation. *Bioinformatics*, 34 (17):2973–2981, 2018.

Kirk Roberts, Bryan Rink, and Sanda Harabagiu. Extraction of medical concepts, assertions, and relations from discharge summaries for the fourth i2b2/va shared task. In *Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data. Boston, MA, USA: i2b2*, 2010.

Kirk Roberts, Dina Demner-Fushman, and Joseph M Tonning. Overview of the TAC 2017 Adverse Reaction Extraction from Drug Labels Track. In *TAC*, 2017.

Frank Rosenblatt. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.

W. D. Ross. Aristotle's metaphysics. a revised text with introduction and commentary. *Mind*, 34(135):351–361, 1925.

Dan Roth and Wen-tau Yih. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL) at HLT-NAACL*, pages 1–8. Association for Computational Linguistics, 2004.

David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

Rune Sætre, Kenji Sagae, and Jun'ichi Tsujii. Syntactic features for protein-protein interaction extraction. In Christopher J. O. Baker and Jian Su, editors, *Short Paper Proceedings of the 2nd International Symposium on Languages in Biology and Medicine (LBM)*, volume 319 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

Sunil Sahu, Ashish Anand, Krishnadev Oruganty, and Mahanandeeshwar Gattu. Relation extraction from clinical texts using domain invariant convolutional neural network. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 206–215. Association for Computational Linguistics, 2016.

Sunil Kumar Sahu and Ashish Anand. Drug-drug interaction extraction from biomedical texts using long short-term memory network. *Journal of Biomedical Informatics*, 86:15–24, 2018.

Sunil Kumar Sahu, Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. Inter-sentence relation extraction with document-level graph convolutional neural network. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4309–4316. Association for Computational Linguistics, 2019.

Victor Sanh, Thomas Wolf, and Sebastian Ruder. A hierarchical multi-task approach for learning embeddings from semantic tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6949–6956, 2019.

Cicero D Santos and Bianca Zadrozny. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1818–1826, 2014.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

Isabel Segura-Bedmar, Paloma Martínez, and María Herrero-Zazo. SemEval-2013 task 9 : Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval)*, pages 341–350. Association for Computational Linguistics, 2013.

Neha Sharma, Vibhor Jain, and Anju Mishra. An analysis of convolutional neural networks for image classification. *Procedia Computer Science*, 132:377 – 384, 2018. ISSN 1877-0509. International Conference on Computational Intelligence and Data Science.

Heng She, Bin Wu, Bai Wang, and Renjun Chi. Distant supervision for relation extraction with hierarchical attention and entity descriptions. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

Yatian Shen and Xuanjing Huang. Attention-based convolutional neural network for semantic relation extraction. In *Proceedings of COLING, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2526–2536. The COLING Organizing Committee, 2016.

Gaurav Singh and Parminder Bhatia. Relation extraction using explicit context conditioning. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1442–1447. Association for Computational Linguistics, 2019.

Alisa Smirnova and Philippe Cudré-Mauroux. Relation extraction using distant supervision: A survey. *ACM Computing Surveys*, 51(5):106:1–106:35, 2019. ISSN 0360-0300.

Steven W Smith et al. The scientist and engineer's guide to digital signal processing. 1997.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the Joint conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211. Association for Computational Linguistics, 2012.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934, 2013.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. N-ary Relation Extraction using Graph-State LSTM. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2226–2235. Association for Computational Linguistics, 2018.

Daniil Sorokin and Iryna Gurevych. Context-aware representations for knowledge base relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1784–1789. Association for Computational Linguistics, 2017.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, pages 2377–2385. Curran Associates, Inc., 2015.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. Supervised open information extraction. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, Volume 1 (Long Papers), pages 885–895. Association for Computational Linguistics, 2018.

Lucas Sterckx, Thomas Demeester, Johannes Deleu, and Chris Develder. Using active learning and semantic clustering for noise reduction in distant supervision. In *Workshop on Automated Base Construction at NIPS (AKBC)*, pages 1–6, 2014.

Ang Sun and Ralph Grishman. Active learning for relation type extension with local and global data views. In *Proceedings of the 21st ACM international Conference on Information and Knowledge Management*, pages 1105–1112. ACM, 2012.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. Multi-instance multi-label learning for relation extraction. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465. Association for Computational Linguistics, 2012.

DR Swanson. Literature-based discovery? The very idea. In *Literature-based discovery*, pages 3–11. Springer, 2008.

Olivier Taboureau, Sonny Kim Nielsen, Karine Audouze, Nils Weinhold, Daniel Edsgärd, Francisco S Roque, Irene Kouskoumvekaki, Alina Bora, Ramona Curpan, Thomas Skøt Jensen, et al. ChemProt: a disease chemical biology database. *Nucleic Acids Research*, 39(suppl_1):D367–D372, 2010.

Yuki Tagawa, Motoki Taniguchi, Yasuhide Miura, Tomoki Taniguchi, Tomoko Ohkuma, Takayuki Yamamoto, and Keiichi Nemoto. Relation prediction for unseen-entities using entity-word graphs. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 11–16, 2019.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory 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: Long Papers)*, pages 1556–1566. Association for Computational Linguistics, 2015.

Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th Annual Meeting*

of the Association for Computational Linguistics (Volume 1: Long Papers), pages 721–729. Association for Computational Linguistics, 2012.

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

Trias Thireou and Martin Reczko. Bidirectional long short-term memory networks for predicting the subcellular localization of eukaryotic proteins. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(3):441–446, 2007.

Paul Thompson, Sophia Daikou, Kenju Ueno, Riza Batista-Navarro, Jun'ichi Tsujii, and Sophia Ananiadou. Annotation and detection of drug effects in text for pharmacovigilance. *Journal of Cheminformatics*, 10(1):37, 2018.

Caroline F Thorn, Teri E Klein, and Russ B Altman. Pharmgkb: the pharmacogenomics knowledge base. In *Pharmacogenomics*, pages 311–320. Springer, 2013.

James Thorne and Andreas Vlachos. Automated fact checking: Task formulations, methods and future directions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3346–3359. Association for Computational Linguistics, August 2018.

Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in the twenty-ninth Annual Conference on Neural Information Processing Systems*, volume 5, 2015.

Bayu Distiawan Trisedya, Gerhard Weikum, Jianzhong Qi, and Rui Zhang. Neural relation extraction for knowledge base enrichment. In *57th Annual Meeting of the Association for Computational Linguistics*, pages 229–240. Association for Computational Linguistics, 2019.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proceedings of the 33nd International Conference on Machine Learning, ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2071–2080. JMLR.org, 2016.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556, 2011.

Vladimir N Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999.

Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. RESIDE: Improving distantly-supervised neural relation extraction using side information. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1257–1266. Association for Computational Linguistics, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, pages 5998–6008. Curran Associates, Inc., 2017.

Patrick Verga, Emma Strubell, and Andrew McCallum. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1)*, pages 872–884. Association for Computational Linguistics, 2018.

FR Vogenberg, C Isaacson Barash, and Pursel M. Personalized medicine: part 1: Evolution and development into theranostics. *PT: A peer-reviewed Journal for Formulary Management*, 35:560–576, 2010.

Ngoc Thang Vu, Heike Adel, Pankaj Gupta, and Hinrich Schütze. Combining recurrent and convolutional neural networks for relation classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 534–539. Association for Computational Linguistics, 2016.

Vishnu Vyas, Patrick Pantel, and Eric Crestan. Helping editors choose better seed sets for entity set expansion. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM, pages 225–234. ACM, 2009. ISBN 978-1-60558-512-3.

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

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: Long Papers)*, pages 1298–1307. Association for Computational Linguistics, 2016.

Wei Wang, Xi Yang, Canqun Yang, Xiaowei Guo, Xiang Zhang, and Chengkun Wu. Dependency-based long short term memory network for drug-drug interaction extraction. *BMC Bioinformatics*, 18(16):578, 2017. ISSN 1471-2105.

Xiaoyan Wang, George Hripcsak, Marianthi Markatou, and Carol Friedman. Active computerized pharmacovigilance using natural language processing, statistics, and electronic health records: a feasibility study. *Journal of the American Medical Informatics Association*, 16(3):328–337, 2009.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes, 2014.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 11–20. Association for Computational Linguistics, 2019.

David S Wishart, Craig Knox, An Chi Guo, Dean Cheng, Savita Shrivastava, Dan Tzur, Bijaya Gautam, and Murtaza Hassanali. DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Research*, 36(suppl_1):D901–D906, 2007.

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.

Ye Wu, Ruibang Luo, Henry C. M. Leung, Hing-Fung Ting, and Tak-Wah Lam. Renet: A deep learning approach for extracting gene-disease associations from literature. In

Lenore J Cowen, editor, *Research in Computational Molecular Biology*, pages 272–284, Cham, 2019. Springer International Publishing. ISBN 978-3-030-17083-7.

Feiyu Xu, Hans Uszkoreit, and Hong Li. A seed-driven bottom-up machine learning framework for extracting relations of various complexity. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 584–591. Association for Computational Linguistics, 2007.

Jun Xu, Hee-Jin Lee, Zongcheng Ji, Jingqi Wang, Qiang Wei, and Hua Xu. UTH_CCB System for Adverse Drug Reaction Extraction from Drug Labels at TAC-ADR 2017. In *TAC*, 2017.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. Semantic relation classification via convolutional neural networks with simple negative sampling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 536–540. Association for Computational Linguistics, 2015a.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. Semantic relation classification via convolutional neural networks with simple negative sampling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 536–540. Association for Computational Linguistics, 2015b.

Peng Xu and Denilson Barbosa. Connecting language and knowledge with heterogeneous representations for Neural Relation Extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3201–3206. Association for Computational Linguistics, 2019.

Wei Xu, Raphael Hoffmann, Le Zhao, and Ralph Grishman. Filling knowledge base gaps for distant supervision of relation extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 665–670. Association for Computational Linguistics, 2013.

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 Conference on Empirical Methods in Natural Language Processing*, pages 1785–1794. Association for Computational Linguistics, 2015c.

Yan Xu, Ran Jia, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, and Zhi Jin. Improved relation classification by deep recurrent neural networks with data augmentation. In *Proceedings of COLING, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1461–1470. The COLING Organizing Committee, 2016.

Yulan Yan, Naoaki Okazaki, Yutaka Matsuo, Zhenglu Yang, and Mitsuru Ishizuka. Unsupervised relation extraction by mining Wikipedia texts using information from the web. 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*, pages 1021–1029. Association for Computational Linguistics, 2009.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems*, pages 5753–5763. Curran Associates, Inc., 2019.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. Structured relation discovery using generative models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466. Association for Computational Linguistics, 2011.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777. Association for Computational Linguistics, 2019.

Hai Ye, Wenhan Chao, Zhunchen Luo, and Zhoujun Li. Jointly extracting relations with class ties via effective deep ranking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1810–1820. Association for Computational Linguistics, 2017.

Zibo Yi, Shasha Li, Jie Yu, Yusong Tan, Qingbo Wu, Hong Yuan, and Ting Wang. Drug-drug interaction extraction via recurrent neural network with multiple attention layers. In Gao Cong, Wen-Chih Peng, Wei Emma Zhang, Chengliang Li, and Aixin Sun, editors, *Advanced Data Mining and Applications*, pages 554–566, Cham, 2017. Springer International Publishing. ISBN 978-3-319-69179-4.

Wenpeng Yin, Yadollah Yaghoobzadeh, and Hinrich Schütze. Recurrent one-hop predictions for reasoning over knowledge graphs. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2369–2378. Association for Computational Linguistics, 2018.

Matthew D Zeiler. ADADELTA: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

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

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

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

Daojian Zeng, Junxin Zeng, and Yuan Dai. Using cost-sensitive ranking loss to improve distant supervised relation extraction. In Maosong Sun, Xiaojie Wang, Baobao Chang, and Deyi Xiong, editors, *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 184–196, Cham, 2017a. Springer International Publishing. ISBN 978-3-319-69005-6.

Wenyuan Zeng, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Incorporating Relation paths in neural relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1768–1777. Association for Computational Linguistics, 2017b.

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

Hong-Tao Zhang, Min-Lie Huang, and Xiao-Yan Zhu. A unified active learning framework for biomedical relation extraction. *Journal of Computer Science and Technology*, 27(6):1302–1313, 2012.

Min Zhang, Jie Zhang, and Jian Su. Exploring syntactic features for relation extraction using a convolution tree kernel. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 288–295. Association for Computational Linguistics, 2006a.

Min Zhang, Jie Zhang, Jian Su, and GuoDong Zhou. A composite kernel to extract relations between entities with both flat and structured features. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 825–832. Association for Computational Linguistics, 2006b.

Ningyu Zhang, Shumin Deng, Zhanlin Sun, Guanying Wang, Xi Chen, Wei Zhang, and Huajun Chen. Long-tail relation extraction via knowledge graph embeddings and graph convolution networks. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3016–3025. Association for Computational Linguistics, 2019a.

Qianqian Zhang, Mengdong Chen, and Lianzhong Liu. A review on entity relation extraction. In *International Conference on Mechanical, Control and Computer Engineering*, pages 178–183. IEEE, 2017a.

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.

Yijia Zhang, Hongfei Lin, Zhihao Yang, Jian Wang, Shaowu Zhang, Yuanyuan Sun, and Liang Yang. A hybrid model based on neural networks for biomedical relation extraction. *Journal of Biomedical Informatics*, 81:83 – 92, 2018a. ISSN 1532-0464.

Yijia Zhang, Hongfei Lin, Zhihao Yang, Jian Wang, and Yuanyuan Sun. Chemical–protein interaction extraction via contextualized word representations and multihead attention. *Database*, 2019b.

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

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

Yuhao Zhang, Peng Qi, and Christopher D. Manning. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215. Association for Computational Linguistics, 2018b.

Zhu Zhang. Weakly-supervised relation classification for information extraction. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, pages 581–588. ACM, 2004. ISBN 1-58113-874-1.

Junzhe Zhao, Tianying Zhou, and Wenhua Dai. Convolutional neural network-based joint extraction of Adverse Drug Events. In *International Conference on Computer Science & Education (ICCSE)*, pages 1–5. IEEE, 2018.

Shubin Zhao and Ralph Grishman. Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 419–426. Association for Computational Linguistics, 2005.

Yi Zhao, Huaiyu Wan, Jianwei Gao, and Youfang Lin. Improving relation classification by entity pair graph. In *Asian Conference on Machine Learning*, pages 1156–1171, 2019.

Wei Zheng, Hongfei Lin, Ling Luo, Zhehuan Zhao, Zhengguang Li, Yijia Zhang, Zhihao Yang, and Jian Wang. An attention-based effective neural model for drug-drug interactions extraction. *BMC Bioinformatics*, 18(1):445, 2017. ISSN 1471-2105.

Wei Zheng, Hongfei Lin, Zhiheng Li, Xiaoxia Liu, Zhengguang Li, Bo Xu, Yijia Zhang, Zhihao Yang, and Jian Wang. An effective neural model extracting document level chemical-induced disease relations from biomedical literature. *Journal of Biomedical Informatics*, 83:1–9, 2018.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 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.

Huiwei Zhou, Huijie Deng, Long Chen, Yunlong Yang, Chen Jia, and Degen Huang. Exploiting syntactic and semantics information for chemical–disease relation extraction. *Database*, 2016a.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212. Association for Computational Linguistics, 2016b.

Peng Zhou, Suncong Zheng, Jiaming Xu, Zhenyu Qi, Hongyun Bao, and Bo Xu. Joint extraction of multiple relations and entities by using a hybrid neural network. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 135–146. Springer, 2017.

Peng Zhou, Jiaming Xu, Zhenyu Qi, Hongyun Bao, Zhineng Chen, and Bo Xu. Distant supervision for relation extraction with hierarchical selective attention. *Neural Networks*, 108:240–247, 2018.

Hao Zhu, Yankai Lin, Zhiyuan Liu, Jie Fu, Tat-Seng Chua, and Maosong Sun. Graph neural networks with generated parameters for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1331–1339. Association for Computational Linguistics, 2019.