Relation Classification with Limited Supervision

Abiola Victor Obamuyide

Department of Computer Science
University of Sheffield

This dissertation is submitted for the degree of
Doctor of Philosophy

June 2020

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Abiola Victor Obamuyide
June 2020

Acknowledgements

I give all glory to God for the blessings of life, provision, and amazing grace, without which nothing would have been possible in the first place.

I would like to express my profound gratitude and appreciation to my supervisor, Dr Andreas Vlachos, for giving me the opportunity to study for a PhD, and for his guidance, support and motivation throughout the PhD journey. He gave me many opportunities, and provided invaluable insights and ideas which helped shape my development as researcher. I could not have wished for a better supervisor and mentor. I am also very grateful to other members of my supervisory team, Dr Mark Stevenson and Professor Georg Struth, for their constructive feedback and suggestions which helped form my research direction. I am extremely grateful to my examiners, Dr Nikolaos Aletras and Prof Danushka Bollegala, for providing very valuable and detailed feedback that helped improve the quality of this thesis.

I would like to thank the SUMMA Project [1] and all the partners involved. Special thanks to Sebastian Riedel and Jeff Mitchel for hosting me at various times, Thomas Demeester, Tim Rocktäschel, Lucas Sterckx and everyone I met at the UCL NLP group for enlightening research discussions; and Afonso, Sebastião, David, Pedro, Mariana, Rui and everyone I met at Priberam for a memorable stay in Lisbon. Thank you all for a great project.

I would like to express my sincere appreciation to all colleagues in Sheffield's Natural Language Processing Group. I thank Mark, Makis, James, Hardy, Ignatius, Zeerak and Tope for their friendship, support and for being awesome lab mates. I am also grateful to other PhDs, Post-docs, Staff and Visitors for various discussions, ideas and assistance.

Special thanks to all Christian friends in the City Life Church family, and especially brothers Ayo, Femi and their families, for being a blessing, and a continual source of spiritual support and encouragement. Thank you for all you do, and may God continue to bless you abundantly in all your endeavours.

[1] This work was supported by the European Union's Horizon 2020 research and innovation project SUMMA (http://summa-project.eu/) under grant agreement No 688139.

I will also like to appreciate many of my pre-Sheffield colleagues and friends for sending greetings and for continued relationships.

Last but not least, my deepest gratitude to my parents, brothers, sisters and my extended family for providing emotional and moral support.

Abstract

Large reams of unstructured data, for instance in form textual document collections containing entities and relations, exist in many domains. The process of deriving valuable domain insights and intelligence from such documents collections usually involves the extraction of information such as the relations between the entities in such collections. Relation classification is the task of detecting relations between entities. Supervised machine learning models, which have become the tool of choice for relation classification, require substantial quantities of annotated data for each relation in order to perform optimally. For many domains, such quantities of annotated data for relations may not be readily available, and manually curating such annotations may not be practical due to time and cost constraints.

In this work, we develop both model-specific and model-agnostic approaches for relation classification with limited supervision. We start by proposing an approach for learning embeddings for contextual surface patterns, which are the set of surface patterns associated with entity pairs across a text corpus, to provide additional supervision signals for relation classification with limited supervision. We find that this approach improves classification performance on relations with limited supervision instances. However, this initial approach assumes the availability of at least one annotated instance per relation during training. In order to address this limitation, we propose an approach which formulates the task of relation classification as that of textual entailment. This reformulation allows us to use the textual descriptions of relations to classify their instances. It also allows us to utilize existing textual entailment datasets and models to classify relations with zero supervision instances.

The two methods proposed previously rely on the use of specific model architectures for relation classification. Since a wide variety of models have been proposed for relation classification in the literature, a more general approach is thus desirable. We subsequently propose our first model-agnostic meta-learning algorithm for relation classification with limited supervision.

This algorithm is applicable to any gradient-optimized relation classification model. We show that the proposed approach improves the predictive performance of two existing relation classification models when supervision for relations is limited. Next, because all the approaches we have proposed so far assume the availability of all supervision needed for classifying relations prior to model training, they are unable to handle the case when new supervision for relations becomes available after training. Such new supervision may need to be incorporated into the model to enable it classify new relations or to improve its performance on existing relations. Our last approach addresses this short-coming. We propose a model-agnostic algorithm which enables relation classification models to learn continually from new supervision as it becomes available, while doing so in a data-efficient manner and without forgetting knowledge of previous relations.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The arrival of the information age has made available a deluge of information in various forms, including audio, video and text. In particular, the World Wide Web (WWW) has further accelerated the rate of growth of this information, as well as providing a scalable and cost-effective way of accessing and utilizing this information. For instance, it has been estimated that the amount of digital data available to us will grow by about 50 folds from the year 2010 to 2020 (Gantz et al., 2012), resulting in an exponential increase in the volume of information within this period. It is desirable to be able to store this information in a form that is maximally useful, available and easy for both man and machine to reason with in the process of solving problems.

This is one of the main motivations for constructing repositories of knowledge such as Knowledge Bases (KB). A KB provides a structured format for organizing, accessing and utilizing information on a large scale. Knowledge bases contain factual information about various entities, their attributes and the relationships between them. The knowledge contained in a KB is stored in such a way as to make machines able to efficiently and effectively read, write, update and reason over such knowledge. KBs typically store information in the form of Resource Description Framework(RDF) [1] triples, which consist of a subject entity, a predicate , and an object entity or attribute value.

In recent years, several efforts have been made in the construction of large knowledge bases, through various community-driven and academic efforts. Examples of such knowledge bases include Freebase (Bollacker et al., 2008a) , NELL (Carlson et al., 2010), YAGO (Suchanek et al., 2007), DBPedia

[1] www.w3.org/TR/2004/REC-rdf-concepts-2004-0210

(Auer et al., 2007) and Wikidata (Vrandečić, 2012). In industry, knowledge bases serve as useful repositories of knowledge that can be easily queried and power various modern day applications, for instance web search and smart personal assistants such as Google Now, Apple's Siri, Microsoft's Cortana, and Amazon's Alexa. Examples of industrial knowledge bases include Google's Knowledge Graph, Microsoft's Satori and Facebook's Entity Graph.

Although knowledge bases can be quite large and contain many facts, entities and relations, they are frequently incomplete. Thus, the facts contained in knowledge bases needs to be updated from time to time. Moreover, because entities are constantly evolving in their roles, associations and attributes, real-life KBs frequently need to be augmented with new relations to extend their coverage.

However, a major challenge that arises when extending knowledge bases to new relations is the lack of sufficient quantities of annotated training data with which to train classification models for the new relations. State-of-the-art approaches for detecting relations between entities are supervised models which require sufficient quantities of supervision to achieve good performance. While sufficient quantities of labelled training data is needed to learn good models for relation classification, such quantities of labelled training data may not be readily available, and even when available, may be expensive and time-consuming to create. Thus, there is the need for the development of approaches for relation classification both when there is zero annotated data, and when there is only limited annotated data for relations. This is the major problem we address in this thesis.

Relation classification aims to detect relations between various entities from text. It is an essential part of many knowledge base population approaches (Ji and Grishman, 2011), and can be useful for a variety of other language processing systems, such as question answering, information retrieval and conversational artificial intelligence systems. A variety of approaches and models have been proposed for the task in the literature, ranging from unsupervised methods which primarily employ clustering-based techniques to semi-supervised methods which provide some guidance in the form of seed instances, to various fully supervised methods, including those that employ manual feature engineering, kernel methods, and deep representation learning methods.

1.2 Aims

Our overall aim in this work is to develop novel approaches for relation classification from text that do not rely on extensive amounts of labelled data, and instead are able to utilize zero or little supervision data for classifying relations. In order to achieve this, we address the following challenges faced in relation classification with zero and limited supervision:

- Existing approaches for relation classification with limited supervision (Rocktäschel et al., 2015; Demeester et al., 2016) propose combining matrix factorization with propositional logic rules. Rocktäschel et al. (2015) assumed that there is sufficient initial supervision for relations that can be used to extract such rules, which may not always be the case especially for new relations with few supervision instances. Demeester et al. (2016) assumed the existence of a relevant external resource that can be used to generate such rules as a source of additional supervision to improve performance for new relations. However, relevant external resources are not always available for all domains, can be incomplete and sometimes contain outdated information, thus rendering any supervision obtained from them unreliable. In Chapter 3 we propose an alternative approach to the use of logic rules when supervision is limited, by investigating how information which is already present within text corpora can be utilized effectively to provide additional supervision for relation classification with limited supervision.

- Many knowledge bases have text descriptions of the relations included in their relation ontology. These descriptions can be regarded as giving a definition to the relations. However, previously proposed approaches to relation classification with limited supervision have mostly ignored this information. The challenge with utilizing such descriptions lies chiefly in how to incorporate them into relation classification systems in such a way as to allow for generalization to the wide range of lexical variations used in expressing relations in text. The work reported in Chapter 4 investigates how relation descriptions can be used to classify relations when no annotated training data is available.

- While a great variety of supervised models have been proposed for the task of relation classification, the majority of current state-of-the-art models are based on the use of neural networks, which requires substantial quantities of supervision data for each relation. In addition, these approaches to relation classification do not have any explicit objective that encourages the models to share and exploit knowledge among all relations. In a limited supervision setting, it should be useful if models are encouraged to exploit knowledge of how to classify one relation to improve performance on other relations. In Chapter 5 we investigate how to reduce the quantity of annotated data required to train supervised neural relation classification models.

- Current approaches for training relation classification models assume the availability of all supervision data for relations in advance before training the model. In order to incorporate newly available supervision into such models after training, either to improve performance on existing relations or to enable classification of new relations, they usually need to undergo substantial retraining. This can be expensive and may lead to the model forgetting how to classify old relations. There is thus the need for models that are able to utilize new supervision as it becomes available, without the need for substantial retraining. Such new supervision can either be used to improve performance on old relations, or to acquire the ability to classify new relations, or both. In Chapter 6 we investigate how to develop relation classification models that are able to learn continually from new supervision as it becomes available, while being data efficient and not forgetting knowledge from previously seen relations in the process.

1.3 Contributions

The contributions of this thesis include the following:

Contextual Pattern Embeddings for Relation Classification with Limited Supervision

While previous work proposed the use of rules mined from external sources, we propose the modelling of contextual surface patterns and their interactions within a Factorization Machines (Rendle, 2010) model for relation classification in limited supervision settings. Contextual patterns, which are the set of surface patterns that are associated with entity tuples in text, are readily

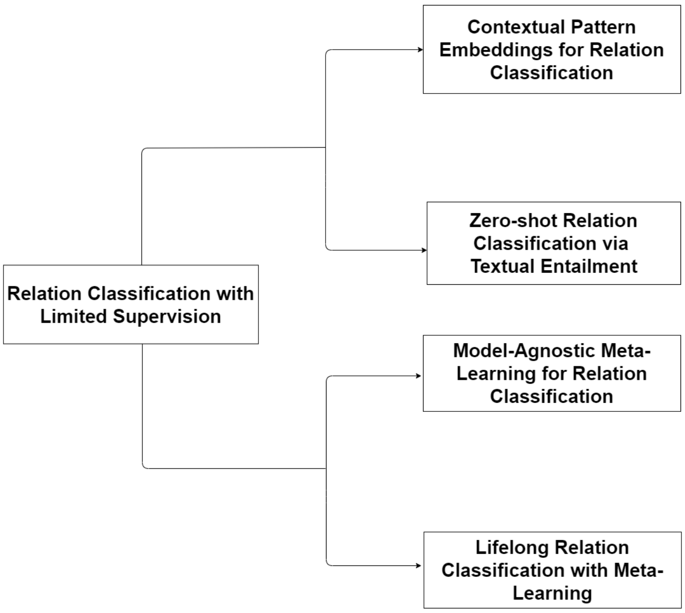

Relation Classification with Limited Supervision
Contextual Pattern Embeddings for Relation Classification
Zero-shot Relation Classification via Textual Entailment
Model-Agnostic Meta-Learning for Relation Classification
Lifelong Relation Classification with Meta-Learning

Figure 1.1 Summary of our Contributions

available within any text corpus, and have the advantage that they do not require consulting external sources to obtain. We investigate learning embeddings for these contextual patterns within a Factorization Machines model for relation classification. We demonstrate that by explicitly modelling the correlations between knowledge base relations and contextual surface patterns we achieve performance equivalent to matrix factorization combined with propositional rules, despite not using such additional supervision in our approach.

Zero-shot Relation Classification as Textual Entailment

We propose an approach and a model for utilizing textual entailment (Fyodorov et al., 2000; Condoravdi et al., 2003; Bos and Markert, 2005; Dagan et al., 2005; MacCartney and Manning, 2009) for relation classification without labelled data. In this formulation, sentences containing at least two entities of interest can be thought of as the premise, and the textual description of the relation of interest as the hypothesis. We show that this formulation leads to several advantages, including the ability to perform zero-shot relation classification by exploiting relation descriptions, use existing textual entailment models for relation classification, and utilize readily available textual entailment datasets to enhance the performance of relation classification systems.

Model-Agnostic Meta-Learning for Relation Classification

We propose to consider the task of relation classification as an instance of meta-learning (Schmidhuber, 1987; Naik and Mammone, 1992; Thrun and Pratt, 1998), and develop a model-agnostic meta-learning protocol for training relation classifiers to achieve enhanced predictive performance in limited supervision settings. This enables us to explicitly optimize the parameters of relation classification models during training for enhanced performance on all relations with limited supervision. We demonstrate that the proposed meta-learning approach improves the predictive performance of two state-of-the-art supervised relation classification models.

Lifelong Relation Classification with Meta-Learning

While most existing relation classification models assume that all supervision data needed for learning is available at training time and are unable to adapt to exploit newly available supervision data to classify new relations without substantial retraining, we propose an approach to make relation classification models able to learn continually by incorporating supervision for new relations as it becomes available, without the requirement for substantial retraining and without forgetting knowledge from past relations, based on a combination of ideas from lifelong learning (Ring, 1994; Thrun, 1996; Zhao and Schmidhuber, 1996) and optimization-based meta-learning.

1.4 Thesis Structure

This rest of the thesis is organized as follows:

Chapter 2 begins by reviewing the task of relation classification and approaches that have been proposed for it in the literature. It also discusses various metrics used for evaluating relation classification systems.

Chapter 3 presents an approach for learning contextual patterns embeddings within a factorization machine framework for relation classification with limited supervision. It shows that effectively modelling the interactions among contextual surface patterns and relations can perform better than combining matrix factorization with propositional logic rules for relation classification with limited supervision.

Chapter 4 describes a principled way to incorporate relation descriptions to enable the classification of relations with zero supervision labels, by formulating the task as one of textual entailment. It demonstrates that this approach performs well on two relation classification benchmarks.

Chapter 5 proposes a model-agnostic algorithm with a training objective that explicitly encourages relation classification models to learn to perform well on all relations, especially when there is only limited supervision. It shows that when evaluated on two datasets, the approach improves predictive performance of two state-of-the-art relation classification models in the limited supervision setting.

Chapter 6 proposes an approach to enable relation classification models exploit newly available supervision to classify new relations continually and efficiently, without forgetting previously learned relations. It reports the results of experiments conducted on two lifelong relation classification benchmarks which demonstrate the effectiveness of the proposed approach in both limited supervision and full supervision settings.

Finally, **Chapter 7** concludes with a discussion of our reseacrh contributions and possible directions of future work.

1.5 Publications

This thesis is based on work reported in the following publications:

1. **Model-Agnostic Meta-Learning for Relation Classification with Limited Supervision**
   Abiola Obamuyide and Andreas Vlachos
   Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019), Florence, Italy.
   Association for Computational Linguistics.

2. **Meta-Learning Improves Lifelong Relation Extraction**
   Abiola Obamuyide and Andreas Vlachos
   Proceedings of the ACL 2019 Workshop on Representation Learning for NLP (RepL4NLP), Florence, Italy.
   Association for Computational Linguistics.

3. **Zero-shot Relation Classification as Textual Entailment**
   Abiola Obamuyide and Andreas Vlachos
   Proceedings of the EMNLP 2018 Workshop on Fact Extraction and VERification (FEVER),Brussels, Belgium.
   Association for Computational Linguistics.

4. **Contextual Pattern Embeddings for One-shot Relation Extraction**
   Abiola Obamuyide and Andreas Vlachos
   Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017) Workshop on Automated Knowledge Base Construction (AKBC), Long Beach, CA, USA.

Chapter 2

Related Work

In this chapter we present an overview of related work in relation classification. We define the task of relation classification, and discuss various approaches that have been proposed for it and the representative methods of each approach. We also elucidate the drawbacks associated with each approach and solutions that have been proposed in the literature, and then conclude with a definition of the evaluation metrics used to measure the performance of relation classification systems.

2.1 Introduction

Relation classification is the task of identifying the semantic relationships between two or more arguments, which are usually entities (Culotta et al., 2006; Bach and Badaskar, 2007). It is an important task for information extraction. In this thesis, we assume that there are two candidate entities involved in each relation, and further that the entities have been identified in a preprocessing stage. This is a common setup for extracting relationships, and is sometimes referred to as just relation extraction in the literature. Examples of semantic relations which can exist between two entities include "`spouse of`" relation between Barack and Michelle Obama, "`founder of`" relation between Steve Jobs and Apple, and "`capital of`" relation between Paris and France.

The output of relation classification can be used to populate an empty *knowledge base* (KB) of relations, or to augment existing knowledge bases such as DBPedia (Auer et al., 2007), Freebase (Bollacker et al., 2008a), YAGO (Suchanek et al., 2007) or Wikidata (Vrandečić, 2012). A relation schema specifies how the information in a KB is structured, including the set of allowable entities and relation types. For instance, the schema might specify that the *spouse of* relation can only hold exclusively between two entities of a certain type (e.g. persons); that the *capital of* relation can only hold between two geopolitical entities which have a geographical location attribute, etc. Thus, a knowledge base contains structured information about entities and their relationships, arranged according to a predefined schema. The task of identifying and labelling the spans of text which are entities is known as named entity recognition (Grishman, Ralph Sundheim, 1996), and is usually required before relation classification. The entire pipeline, from entity identification, coreference resolution and entity linking to relation classification, and afterwards updating a knowledge base with the extracted information is usually carried out as part of the process of knowledge base population (Ji and Grishman, 2011).

2.2 Approaches to Relation Classification

A variety of approaches and setups have been proposed and explored in the literature for extracting relations from text (Bach and Badaskar, 2007; Sarawagi, 2008; Konstantinova, 2014; Pawar et al., 2017), and they can be broadly categorized according to the degree of supervision required in each approach. These are (i) Fully Supervised methods (ii) Unsupervised methods (iii) Semi-Supervised Methods, and (iv) Distantly Supervised methods. In addition to these, we also discuss specific methods for identifying relations relevant to this thesis, such as Neural Networks methods and Matrix Factorization methods.

2.2.1 Fully Supervised Relation Classification

Supervised methods for relation classification require the true labels of all supervision instances. Each instance in the training set is annotated with one of a predefined set of relations. The task is often formulated as a multi-class classification task, with each relation corresponding to a class. Usually an additional dummy or negative relation class is also included to represent instances that do not express any of the target relations. Traditionally, these methods are based on the use of features or kernels.

Feature-based Methods

This approach utilizes a set of features derived from each relation instance to make a classification decision on the instance. The performance of these methods is dependent on the feature set used, and a lot of effort is usually spent on feature engineering - finding a set of features with good discriminative performance on the relation instances. These features can be any combination of lexical, syntactic or semantic features of the instances.

Kambhatla (2004) explored the impact of a number of lexical, syntactic and semantic features for extracting relations, and presented results of utilizing such features within a maximum entropy classifier on the Automatic Content Extraction (ACE) (Doddington et al., 2004) dataset. They found that a combination of the features gives the best performance on the dataset. Similarly, a systematic study of the effectiveness of a variety of feature combinations for relation classification was carried out by Jiang and Zhai (2007). They report that while they found each of the lexical, syntactic and semantic features to be effective on their own, syntactic features were generally the most effective of the three, and that simple lexical, syntactic and semantic features generally performs well. They also report some gains in performance when using all the three types of features.

Kernel-based Methods

The performance of feature-based relation classification methods rely largely on the set of features used, thus requiring a lot of manual effort in the design and selection of discriminative features. Instead of explicit feature engineering, kernel-based methods offer the alternative of designing kernel functions which compute similarities between instances efficiently in an implicit, high-dimensional feature space. One of the earliest application of kernel methods for natural language tasks was by Collins and Duffy (2002) who proposed kernels whose computations are performed over parse trees, and Lodhi et al. (2002) who present kernels whose computations are defined over strings of characters (sequences). The kernels proposed by Collins and Duffy (2002) compute similarity of two instances as the number of shared subtrees in their syntactic trees. The kernel utilizes a high-dimensional, implicit feature space where each dimension represents one possible subtree, such that the value computed by the kernel for any two instances represents their inner product in the high-dimensional space. Similarly, the kernel proposed by

Lodhi et al. (2002) compute the similarity of two instances as the number of shared subsequences in their string (sequence) representations.

Kernel-based methods for relation classification represent relation instances as a vector of lexical, syntactic and semantic features. The similarity between two relation instances is then computed in terms of the degree of overlap in their representations. There are different types of kernels that have been proposed for the task of relation classification, including sequence kernels, syntactic tree kernels and composite kernels.

Sequence kernels represent each relation instance as a sequence of items and the kernel function measures the similarity of two instances by computing the extent of overlap of their subsequences. Bunescu and Mooney (2005b) present an extension of the sequence kernels proposed by Lodhi et al. (2002) which is able to work with sequences composed of combinations of words and part of speech tags. Bunescu and Mooney (2005b) represent each component of sequences (i.e each word or PoS tag) with a feature vector representation, thus turning each relation instance into a sequence of vector representations. A Support Vector Machines (Cortes and Vapnik, 1995) classifier is then trained to distinguish positive from negative relation instances.

Syntactic tree kernels define their computations over parse tree representation of relation instances, and measure the similarity of two relation instances by computing the extent of overlap of their subtrees. The parse tree of a relation instance represents the instance in terms of its syntactic constituents, such as verb phrases, noun phrases and the words in the instance. It is also possible to augment the parse tree of instances with additional information such as part of speech tags. Syntactic tree kernels for relation classification are explored in Zelenko et al. (2003), who proposed kernels which compute similarities over the shallow parse tree representations of relation instances, together with algorithms for computing such kernels.

Some tree kernel approaches are based on the use of features derived from the dependency parse tree of an instance, which describes the grammatical relations between the words that make up the sentence. In a dependency tree, the words that make up a sentence form the nodes, with a directed edge from a word (dependent) to its parent. The tree kernels proposed by Zelenko et al. (2003) for parse tree representations was extended by Culotta and Sorensen (2004) to compute the similarity between two given dependency trees. In order to represent each relation instance, Culotta and Sorensen (2004) consider the smallest subtree of each sentence which contains both the subject and object of the relation. In addition, they also augment each node in the dependency tree with additional features such as its PoS tag, entity type, WordNet hypernyms and position.

Bunescu and Mooney (2005a) proposed a shortest path dependency kernel for relation classification, which is based on the idea that the relation between two entities can be determined based on the information along the shortest dependency path between the two entities. Each relation instance is represented as the lexicalized shortest path between the two entity mentions in its dependency tree, and the kernel is designed to compute the similarity between two relation instances from this representation. To prevent data sparsity resulting from the use of just words in the lexicalization of a dependency path, words are replaced by their corresponding word classes, such as part of speech tags and entity types.

It is also possible to combine the information from multiple individual kernels into a single composite kernel. As stated by Pawar et al. (2017), information captured by a sequence kernel can be combined with information captured by a syntactic tree kernel into a single composite kernel, through a sum, linear combination or a product of the two kernels. It is however important to ensure the individual kernels are normalized prior to their combination, in order to have balanced contribution from all kernels in the final result.

Zhao and Grishman (2005) proposed to combine information obtained from sentence tokenisation together with that obtained from parsing for relation classification. They designed different kernels to take advantage of each type of information. They observe that performance progressively increases as they add each of their kernels, since each successive kernel helps to address mistakes made by previously added kernels. They report that a composite kernel made up of a combination of all the kernels results in the best overall performance. Nguyen et al. (2009b) also explored the use of composite kernels to capture various syntactic and semantic information for relation classification. They designed a kernel to capture syntactic information obtained from the constituency and dependency tree of relation instances, and a separate kernel to capture information obtained from entity types and lexical sequences. They combine the kernels into a composite kernel, and report that it outperforms previous systems.

2.2.2 Unsupervised Methods

Unsupervised methods for relation identification aim to be able to discover relations without any labelled data or predefined relation inventory. There are a number of approaches under this paradigm, including clustering-based methods and topic-based methods.

Hasegawa et al. (2004) proposed an early clustering-based approach to unsupervised discovery of relations. Their approach requires only that named-entity mentions must be identified beforehand, for instance by preprocessing the corpus with a named entity tagger. The approach starts by grouping co-occuring entity pairs together and then aggregating their contexts across the corpus together. Based on the aggregated entity pair contexts, similarities between the entity pairs are computed and those with the highest similarity are grouped together in a cluster as belonging to a relation. For a given entity pair, Hasegawa et al. (2004) consider it to be co-occuring if they are not more that N words apart, where N is a parameter which is set empirically. The words in between entities in the pair, aggregated across all its occurrences in a corpus, is taken as the context for the mention pair. Then, a vector of tf-idf counts is computed for each mention pair with its context. To determine the similarity of two pairs of mentions, the cosine product of their context vectors is taken. Mention pairs are then clustered according to their similarity scores using hierarchical clustering.

The Discovery of Inference Rules in Text (DIRT) algorithm was proposed by Lin and Pantel (2001) as a way of inducing relation types by generalizing dependency paths of relation instances. Their proposed algorithm is inspired by the distributional similarity hypothesis (Harris, 1954; Firth, 1957), but instead of identifying similar words, identifies dependency paths which tend to link the same set of words. They compute similarities between dependency paths, and generate inference relationships between the most similar dependency paths. Lewis and Steedman (2013) presented an approach for discovering clusters of relation types in a bilingual setting, using English and French as a case study. Their approach is based on exploiting the alignments between named entities in the two languages to discover text patterns expressing the same relations. Inspired by Latent Dirichlet Allocation (LDA)-based topic models (Blei et al., 2003), Yao et al. (2011) propose a series of methods for discovering clusters of relation triples from a corpus that utilizes generative probabilistic models for modelling entity pairs and the syntactic dependency paths between them. Unlike standard topic models where the observations are words and each word is assumed to be drawn from a topic distribution indicated by a latent topic indicator variable, Yao et al. (2011) assume that a document is made up of an exchangeable collection of relation facts, where each relation fact is drawn from a relation type distribution selected by a latent relation type variable. Their approach further imposes entity type constraints on the induced relations, and exploits features on the dependency path between entity mentions. Building on the work of Yao et al. (2011), De Lacalle and Lapata (2013) propose to automatically integrate general domain knowledge in the form of first order logic rules into topic models. This has the advantage that induced relation clusters respect the contraints imposed by both the data distribution and human-specified domain rules.

An approach based on paraphrase acquisition was proposed for unsupervised relation discovery by Romano et al. (2006). For their approach, they assume a list of lexico-syntactic templates which entail the relations of interest and a syntactic matcher to identify the occurences of the given templates from text. The syntactic matcher is based on a sequence of three transformations, which include syntactic dependency parsing text, then matching the entailing templates in the dependency parse, and finally extracting candidate relation arguments which match the template argument variables. The authors give an example for the template “X interact with Y”, for which their approach produces the paraphrases “X bind to Y”, “X activate Y”, etc.

2.2.3 Semi-Supervised Approaches

While unsupervised relation discovery methods can detect relations without supervision data, the extracted clusters of relations are sometimes incoherent, and mapping the relation clusters into a semantic relation schema afterwards is not always straightforward. However, generating labelled data for fully supervised relation classification can be time consuming and expensive in terms of both cost and the effort required. It is desirable to have relation classification methods that are not dependent on completely annotated training data, thus reducing both the cost and manual effort required for extracting relations. Semi-supervised approaches have been proposed to improve on unsupervised methods by utilizing limited supervision together with a large amount of unlabelled data for extracting relations. We next give an overview of representative methods under this approach.

Bootstrapping Methods

Bootstrapping-based approaches for extracting relations typically require just a few seed instances of each relation, in addition to a large unlabelled corpus of text, such as the Web. The seed instances given to these approaches usually include example subject and objects of each relation. Based on the given initial supervision, these approaches return as output relation instances from the unlabelled corpus. Examples of early boostrapping approaches include the Dual Iterative Pattern Relation Expansion (DIPRE) algorithm (Brin, 1998) and Snowball (Agichtein and Gravano, 2000).

The DIPRE algorithm is based on the *pattern relation duality* principle, which states that given a good set of relational patterns, a good set of entity tuples can be obtained, and conversely, given a good set of entity tuples, a good set of patterns can be obtained. Algorithm 1 below gives a sketch of the DIPRE algorithm.

Algorithm 1 Dual Iterative Pattern Relation Expansion (*DIPRE*)

```
Require: Set of $S$ tuples known to be in relation $R$
Output: Set $S$ grown over multiple iterations

while there are no new tuples to be added do
    Find all occurrences of the tuples from $S$ on the Web
    Learn patterns $P$ from these occurrences
    Search the web using $P$ to find new tuples $T$
    Add $T$ to $S$
end while
```

DIPRE accepts as input seed instances the entities pairs and the patterns between them specified in form of tuples. It then proceeds iteratively, first finding occurrences of the entity pairs on the Web, extracting relation patterns from the found entity pairs, searching the web with the acquired patterns to find new entity pairs, and augmenting the initial set of seed instances with the newly-found instances. The process is repeated until a predefined number of entity pairs have been found by the system. Brin (1998) report that with just three seed examples of the "author" relation, their system was able to generate 15257 unique author and book instances from a corpus of 24 million web pages.

Agichtein and Gravano (2000) propose a system called Snowball that improves the DIPRE boostraping algorithm by generalizing how its relation patterns and tuples are generated, represented and evaluated. Instead of using regular expressions to match entity mentions in the relation patterns as was done in DIPRE, Snowball made use of entity tags instead. This improves the precision of the patterns, since this ensures that selected entities are valid relation arguments. Snowball represents the words surrounding entities with vectors, which enables similarities of entity context to be computed with the inner product between vectors. In addition, Snowball evaluates the precision of discovered patterns in each iteration and discards those which have low precision, as determined by a confidence score which can be computed as the fraction of correct tuples the pattern is able to correctly extract to the total number of tuples retrieved. Tuples can in turn be evaluated based on the confidence values of the patterns that extract them. Thus unlike DIPRE, Snowball is able to filter out low quality patterns and tuples in each of its runs in order to improve the quality of extracted relation facts.

Active Learning based Approaches

A shortcoming common to all bootstrapping-based approaches is that they suffer from *semantic drift* (Riloff and Jones, 1999; Curran et al., 2007), a phenomenon whereby the precision of their extractions gradually deteriorates over time. However, labelling a sufficient number of supervision instances for training supervised systems can be expensive and time consuming. This is one of the main motivations behind the introduction of active learning techniques (Settles, 2010) to select the most useful training instances for annotation, thereby reducing the time and cost of data labelling, under the assumption that the model is allowed to request for the labels of a limited number of unlabelled instances. In order to determine which instances would be selected for annotation from a large pool of unlabelled instances, various criteria have been proposed. The overall objective of all active learning approaches is to be able to achieve performance comparable to that of a fully supervised system with a fraction of the training data.

One of the earliest applications of active learning for extracting relations was by Sun and Grishman (2012), who propose a system called *LGCo-Testing* that uses active learning based on the co-testing (Muslea et al., 2000) selective sampling framework for relation classification. It uses two classifiers based on two different views of the data, a maximum entropy classifier which uses features local to each relation instance (local view), and a k-nearest neighbour classifier which utilizes global features based on the distributional similarities between words occurring in the relation contexts computed from a 2 billion token corpus (global view). The instances selected for annotation are then those with labels on which the global and local classifiers disagreed the most. When compared to a completely supervised system, the authors report that their approach leads to reduction in annotations by up to 97% while maintaining the same level of performance.

Open Information Extraction (OpenIE)

Banko et al. (2007) proposed the concept of open information extraction, which is able to discover potentially interesting relations between entities automatically from a text corpus. They introduced *TextRunner*, an early OpenIE system. *TextRunner* is made up of three core modules, namely the self-supervised learner, the single-pass extractor, and the redundancy-based assessor.

The self-supervised learner heuristically labels extracted entity tuples as either positive or negative candidate relation instances, based on a set of predefined rules. It then represents each candidate relation instance with a vector of features, and trains a Naive Bayes classifier to distinguish the positive from the negative relation instances.

The single-pass extractor is run once over the corpus to annotate it with PoS tags and noun phrase chunks. It considers each pair of noun phrase chunks found within a sentence as a candidate relation instance, and obtains the relation name by heuristically eliminating unimportant phrases from the words occurring between such pairs. It uses the Naive Bayes classifier built in the previous step to classify each relation instance into positive and negative relation instances, and then accepts and stores only the positive relation instances into the system.

The redundancy-based assessor assigns a probability of correctness to each extracted fact tuple, by first automatically merging extracted relation tuples which have the same entities and relation names, while keeping track of the number of sentences from which each extraction was made. It then uses the sentence counts to compute the probability of correctness of each extracted tuple.

Fader et al. (2011) observed that the output of *TextRunner* suffers from a number of shortcomings, which severely affect its quality. These include incoherent extractions and uninformative extractions. Incoherent extractions occur when the extracted relation phrases does not have any meaningful interpretation. This is usually as a result of the word by word sequential decisions made by *TextRunner* on whether or not to include words in the relation phrase. The authors give an example of a sentence that leads to such extractions in TextRunner as “Extendicare agreed to buy Arbor Health Care for about US $432 million in cash and assumed debt.”, for which TextRunner returns an extraction such as (Arbor Health Care, for assumed, debt). The relation phrase “for assumed” is not valid since it starts with a preposition.

Uninformative extractions are those which omit important information from the relation phrase. For instance, from the sentence “Bill Gates made a donation to the UN”, TextRunner may extract (Bill Gates, made, a donation) instead of (Bill Gates, made a donation to, the UN). The authors note that though this problem can be partially addressed with syntactic constraints, this can result in the extraction of overly long relation phrases such as “is offering only modest greenhouse gas reduction targets at” from the sentence “The Obama administration is offering only modest greenhouse gas reduction targets at the conference”.

To address the above short-comings of *TextRunner*, Fader et al. (2011) propose *ReVerb*, an OpenIE system with better quality of extractions.

In order to correct the problem of incoherent extractions, *ReVerb* introduced the use of syntactic constraints on possible relation phrases that can be extracted. One of such constraints is that a relation phrase has to be either a verb, a verb followed by a preposition, or a verb followed by nouns, adjectives or adverbs together with a preposition. If there are adjacent matches within a sentence, they are merged into a single relation phrase. Also, the system requires that the relation phrase must be located between its two arguments within a sentence.

To address the problem of uninformative extractions, ReVerb uses lexical constraints, such as requiring that for a relation phrase to be valid, it must have been observed with at least k relation arguments in the corpus, where k is a hyperparameter set to 20 in ReVerb.

Overall *ReVerb* adopts a different approach to *TextRunner* for extracting relations. In contrast with *TextRunner*, it makes globally informed decisions about relation phrases rather than word-by-word decisions.

2.2.4 Distant Supervision

Distant Supervision is an approach to obtaining noisy labels for training data (relation instances for training relation classification models) from existing semantic knowledge bases, such as Freebase, Wikidata or DBPedia. It is regarded as a method for obtaining *weak* labels for training data since the annotations obtained from it are not always the ground truth labelling of the data. However, it is a popular method as it provides an inexpensive way to provide labels for unlabelled data. One of the earliest proposals to create "weakly labelled" training data from a knowledge base was carried out for the biomedical domain by Craven and Kumlien (1999). Later, similar ideas were proposed by Mintz et al. (2009), Bunescu and Mooney (2007) and Nguyen et al. (2007). Mintz et al. (2009) utilized Freebase as the knowledge base for generating weak labels for various relations and referred to the idea as *"distant supervision"*, a term which is now commonly used to refer to the approach. In order to annotate sentences with relations from the knowledge base, distant supervision utilizes a labelling heuristic known as the *distant supervision assumption*, which can be summarized thus:

> **Distant Supervision Assumption** : If two entities participate in a relation, then any sentence that contains the two entities might express that relation.

As an illustration, given that the entity pair *Barack Obama, United States* is present in the knowledge base for the relation `country of birth`, the following sentences which contain both entities are considered by distant supervision to be instances of the `country of birth` relation:

1. Barack Obama was born in Honolulu, Hawaii, United States
2. Barack Obama is the 44th President of the United States
3. Barack Obama was seen with his wife Michele in Chicago, United States

To train a relation classifier for the `country of birth` relation, all of the above sentences are taken as positive relation instances. Since negative relation instances are also required to train a relation classifier, other sentences which do not contain the entity pairs known to be participating in this relation are taken to be negative instances. After the extraction of features from the training instances, any supervised classifier, such as Logistic Regression may then be trained to classify test relation instances.

As a result of the noisy data annotation, which leads to both false positives in the training data (for instance the second and third sentences above) and false negatives (when a sentence that expresses a relation is omitted from the training data because the entities in the sentence are not present in the knowledge base), distantly supervised relation classification systems suffer from reduced performance compared to relation classification systems trained on manually annotated data. As a result of this, a variety of approaches have been proposed to address the problem of false positives and false negatives. We give a brief overview of these approaches.

Reducing False Positives in Distant Supervision

A number of approaches have been proposed to address the issue of false positives in distant supervison data, and Roth et al. (2013) present a survey of these approaches. The fundamental premise of distant supervision is that all sentences containing entities known to be in a relation in a knowledge base are likely to express that relation. As the example sentences above illustrates, this is not always the case. Hence Riedel et al. (2010) argue that the distant supervision assumption is too strong. They proposed instead the *expressed-at-least-once* assumption, which relax the fundamental distant supervision assumption, requiring only that at-least one of the selected sentences should express the relation of interest.Riedel et al. (2010) then proposed a noise reduction model based on the new assumption. The noise reduction approach of Riedel et al. (2010) utilizes a factor graph which models the relation existing between two entities and the sentences which might express the relation. The model is not given information about which sentences express each relation, but rather the model is penalized whenever it does not satisfy the expressed-at-least-once constraint. They consider the task as an instance of constraint driven semi-supervised learning (Chang et al., 2008), and utilized Gibbs sampling (Geman and Geman, 1984; Jensen et al., 1995) with SampleRank (Wick et al., 2009) for inference.

Following their proposal, a number of other noise reduction approaches have also been proposed (Hoffmann et al., 2011; Surdeanu et al., 2012a). These approaches are typically based on a probabilistic graphical modelling framework with the *expressed-at-least-once* assumption included as a constraint, though they differ with respect to the dependencies assumed to be present in the data, the modelling stage at which the constraint is enforced and the specific graphical model inference method employed.

Hoffmann et al. (2011) introduce *MultiR*, which extends the model of Riedel et al. (2010) to the multi-label setting. *MultiR* is able to model entity pairs which participate in overlapping relations, for instance the pair *(Satya Nadella, Microsoft)* which participates in the relations *employee_of* and *ceo_of* at the same time. To allow predicting more than one relation per entity pair, *MultiR* has a variable for each entity pair and relation that indicates if the relation holds for the entity pair, thus allowing for the prediction of more than one relation for that pair. Values are assigned to the variables via distant supervision during training. The model also contains factors that represent the per-sentence relation predictions and ensure that for each entity pair, only one relation is predicted per sentential context. In other words, while it allows each entity pair to take part in multiple relations, only one such relation can hold for the entity pair in a particular sentence. To estimate model parameters, the authors employ a perceptron-style (Collins, 2002) training algorithm.

Building on the work of Hoffmann et al. (2011), Surdeanu et al. (2012a) proposed the Multi-Instance Multi-Label Relation Extraction (*MIMLRE*) model, a probabilsitic graphical model which makes predictions in two stages. In the first stage, a multi-class relation classifier makes predictions for each sentential context. The predictions from the first stage is then aggregated by a collection of per-relation binary classifiers in the second stage to make predictions for each entity pair. In this model, the expressed-at-least-once constraint is implemented by means of a feature in the per-relation classifiers which fires when the relation has been predicted at least once for the entity pair in the set of its sentential contexts. The authors report results which show that their approach outperforms the previous approaches of Riedel et al. (2010) and Hoffmann et al. (2011).

A number of other noise-reduction approaches are based on the use of surface patterns. A generative, hierarchical topic model for reducing false positives by scoring and filtering relational patterns was proposed by Alfonseca et al. (2012). The model was inspired by that proposed for multi-document summarization in Haghighi and Vanderwende (2009), except that Alfonseca et al. (2012) consider relation patterns as words and entity pairs as documents. In addition, Alfonseca et al. (2012) further divide entity pairs into groups according to which relations they participate in. The generative model assumes three topic distributions : a background topic distribution over patterns common to all relations, an entity pair-specific topic distribution over patterns, and a relation-specific topic distribution which is used to estimate the probability that a surface pattern expresses a relation. The authors employ Gibb's sampling as the inference algorithm. As noted in Roth et al. (2013), the advantage of approaches that use topic models for noise reduction, in contrast to those that use the *expressed-at-least-once-assumption*, is that they do not require separate negative instances for training.

Takamatsu et al. (2012) proposed a different generative model for noise reduction. In contrast to the model of Alfonseca et al. (2012), which utilizes the indirect approach of first modelling the generative process of patterns and then utilizing that to model their probability of expressing a relation, Takamatsu et al. (2012) aim to directly model the probability of a pattern expressing a relation or not. The underlying assumption of their model is that if a pattern matches a relation's entity pair, then either the pattern expresses the relation, or it has a high overlap in entity pairs with other patterns that express the relation, although it does not explicitly rule out the possibility that the entity pairs of a pattern which expresses a relation may still be observed with other patterns that do not express the relation. In addition, unlike the approach of Riedel et al. (2010), the model of Takamatsu et al. (2012) groups relation patterns together for each entity pair, and has the advantage that it is able to handle cases where there is only one mention of an entity pair in the corpus. They evaluate their approach on a dataset derived from Wikipedia and report better performance compared to previous noise reduction approaches.

Reducing False Negatives in Distant Supervision

False negatives are instances which are assumed to not have any relations because they are not present in the knowledge base used for distant supervision. This is often a result of the incompleteness of the knowledge bases used for distant supervision, as most of them are incomplete. Even the largest knowledge bases, such as Freebase and Wikidata are still missing instances of many relations (for instance, Min et al. (2013) report that 93.8% of `persons` in Freebase have no *place of birth* attribute). Consequently, if such false negatives are used as part of distant supervision training data, they would lead to suboptimal performance for relation classification models trained with such data. In order to reduce the impact of false negatives on performance, Min et al. (2013) extend the MIML model of Surdeanu et al. (2012a) to use fewer negative training examples and instead utilize more positive and unlabelled examples. They propose a 4-layer hierarchical graphical model with latent variables that represent the true label assignment of training examples.

To train their model, they utilized hard Expectation Maximization (EM) [1] together with a log-likelihood objective. The authors evaluate their model on the KBP dataset of Ji et al. (2010) and report performance improvements over the approaches proposed in Surdeanu et al. (2012a) and Hoffmann et al. (2011) for reducing noise.

Xu et al. (2013) propose to increase the quantity and quality of supervision data through the use of a pseudo-relevance feedback mechanism. The approach assumes entity pairs which appear more frequently in relevant sentences are likely to express the relation of interest, and augments a knowledge base with missing relation instances extracted by exploiting information from the passage retrieval model of Xu et al. (2011). To increase recall of such entity pairs, the passage retrieval model utilizes coarse features, which they combine with fine features obtained from the system of Hoffmann et al. (2011) to encourage high precision. After relation mentions are annotated with distant supervision, the passage retrieval model, which is a ranker based on Support Vector Machines (Cortes and Vapnik, 1995), is trained on the same dataset to provide relevance feedback on the distantly annotated mentions. The distantly annotated mentions are then filtered with the passage retrieval model, and thereafter used to train the *MultiR* model of Hoffmann et al. (2011). When compared with the same model trained on the unfiltered distantly annotated mentions, they report that their approach leads to improvements in performance.

Some approaches propose to mitigate the impact of false negatives by augmenting distantly annotated data with manually annotated data. Nguyen and Moschitti (2011) explored combining manually labelled data from the ACE (Doddington et al., 2004) dataset with distantly annotated text from Wikipedia. For their experiments, they linearly interpolated the outputs of kernel-based supervised relation models from Zhang et al. (2006) and Nguyen et al. (2009a). They report that the models trained on both distantly-annotated data and manually annotated ACE data outperform that trained on manually annotated ACE data alone.

Pershina et al. (2014) extended the *MIML* model of Surdeanu et al. (2012a) with a new layer containing a set of latent variables to model the human ground truth labels for each instance in distantly annotated relation instances. The human ground truth labels were extracted from the KBP dataset of (Ji

[1] In hard EM, the expectation step computes the most probable value (or assignment) of the latent variable, while in soft EM the expectation step retains a probability distribution over the possible values of the latent variable. See, for instance, MacKay (2003) for more information.

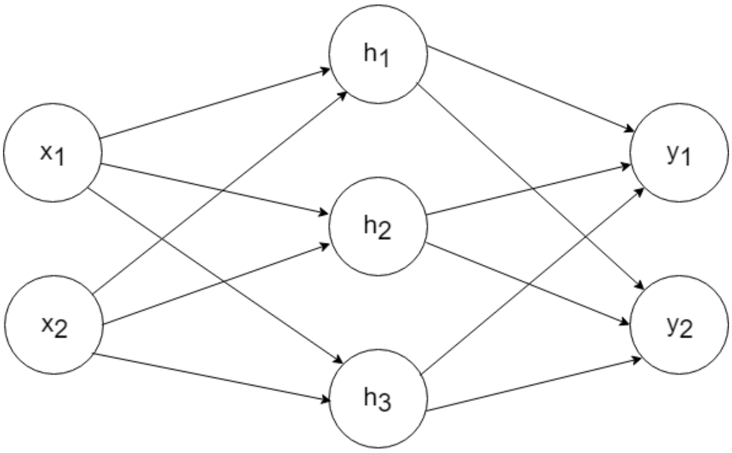


Figure 2.1 Illustration of a fully-connected neural network with one input layer, one hidden layer and one output layer.

and Grishman, 2011) and incorporated into the training process of MIML in the form of "relation guidelines" which encode the entity type preferences and indicative dependency paths of each relation. During training, the labels obtained from such relation guidelines are able to override the labels assigned to relation mentions by the original MIML model. When evaluated on the KBP dataset, the authors report their approach improves performance compared to various previous models such as those proposed in Surdeanu et al. (2012a), Mintz et al. (2009), Hoffmann et al. (2011) and Min et al. (2013).

Angeli et al. (2014) explored the use of various active learning criteria for selecting a limited number of distantly annotated mentions for manual annotation. They combine such manually annotated mentions together with a much larger distantly annotated corpus for relation classification. The authors compared three active learning criteria for selecting instances to annotate: sampling uniformly at random, sampling by high Jensen-Shannon divergence, and a new criterion proposed by the authors, which samples instances which are both uncertain and representative of the other instances . They perform experiments on the 2010 and 2013 versions of the KBP dataset, and report that the use of active learning generally improves over the MIML model of Surdeanu et al. (2012a) trained without active learning, and that their proposed active learning criterion outperforms the other alternatives.

2.2.5 Neural Networks Methods

In this section we first present an overview of neural networks and their various variants. We then present specific neural network architectures and approaches that have been proposed for relation classification.

Artificial Neural Networks, usually referred to as Neural Networks (NNs), are learning models inspired by, and which are a simplification of, how neurons work in biological brains. They are usually made of a set of interconnected nodes, with each node being the artificial analogue of the biological neuron. A node receives input, processes it and then passes on the result of its processing to other nodes connected to it. The edge connecting two nodes is typically associated with a parameter value, which is learnt during training. They can be nested arbitrarily deeply in layers, with each layer computing an arbitrary transformation of its input. Neural networks have the ability to use non-linear activation functions to map layer inputs to outputs, and as a result they are able to learn highly non-linear compositional functions to map inputs to outputs. They are trained by minimizing the empirical loss on data using a gradient-based optimization algorithm such as Stochastic Gradient Descent (Nemirovski and Yudin, 1978). The process of passing inputs through the successive layers of a neural network to produce outputs is known as forward propagation, while the gradient of the loss with respect to the parameters of the network is computed with a procedure termed backpropagation (Rumelhart et al., 1988; Werbos, 1990), which utilizes the chain rule of calculus together with dynamic programming for efficient gradient computations. Figure 2.1 gives an illustration of a feed-forward neural network with a single hidden layer. In order to utilize neural networks for natural language tasks, each word in the input is usually represented by an embedding vector, also known as a *word embedding* vector. Common neural network-based architectures that have been proposed include Recurrent Neural Networks (RNNs) (Jordan, 1986; Elman, 1990), Long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) networks and Convolutional Neural Networks (CNNs)(LeCun et al., 1998).

RNNs can be thought as a flexible variant of feed-forward neural networks. In contrast to feed-forward neural networks which have a predefined number of hidden layers, RNNs are able to change the number of their hidden layers such that each layer of the network corresponds to a new time step of the input and share parameters between the layers. RNNs are made mainly of recursively defined functions which accept as input the current input and previous state, and produce a current state vector, which can optionally be

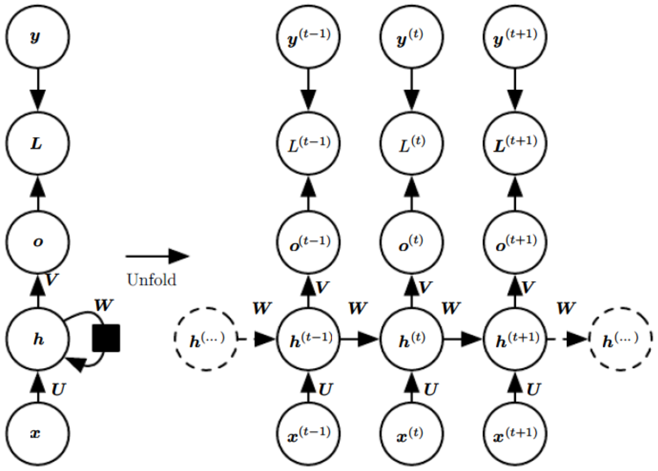


Figure 2.2 An illustration of the computations performed by an RNN model, drawn with recurrent connections (left), and unfolded in time (right). L is a loss function which measures the discrepancy between model outputs $\boldsymbol{o}$ and true target values $\boldsymbol{y}$ at each timestep t (Goodfellow et al., 2016).

mapped to an output vector for the current time step. To encode an input of a certain length, RNNs are unrolled in time to the same length as the input, which enables them to encode inputs of arbitrary length.

As illustrated in Figure 2.2, when given input $\boldsymbol{x}$ at every time step t, a RNN produces hidden representations $\boldsymbol{h}$ and outputs $\boldsymbol{o}$ through the following computations (Goodfellow et al., 2016):

$$\begin{aligned} \boldsymbol{h}^{(t)} &= \tanh\left(\boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)}\right) \\ \boldsymbol{o}^{(t)} &= \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^{(t)} \end{aligned} \tag{2.1}$$

where $\boldsymbol{b}, \boldsymbol{c}$ are bias vectors and U, V, W are parameter matrices.

In practice , the RNN model as defined suffer from the problem of vanishing and exploding gradients (Bengio et al., 1994; Pascanu et al., 2013), where the magnitude of gradients reduce to zero or increase to excessively large values during training as a result of repeated matrix multiplications during the backpropagation process, thus making learning difficult.

LSTMs are a special type of RNNs proposed to address the problem of vanishing and exploding gradients during training. They introduce mechanisms which prevent the underflow and overflow of gradients during the training process. LSTMs introduce a series of gates to decide which information should be kept and which should be discarded during the backpropation process. Specifically, LSTMs are equipped with an input gate $\boldsymbol{i}$, a forget gate

$\boldsymbol{f}$ and an output gate $\boldsymbol{o}$, all of which are dependent on the current input and the previous hidden state for their computations at each time step t. The computations performed by an LSTM model are summarized in the Equations below:

$$\begin{aligned}
\mathbf{i}_t &= \sigma\left(\mathbf{W}_i x_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i\right)\\
\mathbf{o}_t &= \sigma\left(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o\right)\\
\mathbf{f}_t &= \sigma\left(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f\right)\\
\mathbf{c}_t &= \mathbf{i}_t \circ tanh\left(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c\right) + \mathbf{f}_t \circ \mathbf{c}_{t-1}\\
\mathbf{h}_t &= \mathbf{o}_t \circ tanh\left(\mathbf{c}_t\right)
\end{aligned} \tag{2.2}$$

where $\boldsymbol{c}$, $\boldsymbol{h}$ and $\boldsymbol{b}$ are the cell state, output and bias vectors respectively, and U, W are parameter matrices.

A bidirectional LSTM (BiLSTM) variant, which runs separate LSTMs forward and backward over the sequence, was introduced by Graves et al. (2013). The hidden state of each time step is obtained by concatenating the hidden states from the forward and backward LSTMs at each time step. Cho et al. (2014) introduced a slight variation of LSTMs called the Gated Recurrent Unit (GRU). In contrast to LSTMs, the GRU uses a fewer number of gates and no separate memory component, and therefore has the advantage that it requires fewer computational resources.

Relation Classification with Neural Networks

A number of methods for relation classification have been proposed that rely exclusively on the use of various neural network architectures. Socher et al. (2012) proposed the use of Matrix-Vector Recursive Neural Networks (MV-RNN), which learns compositional representation for phrases in a parse tree structure, for relation classification. *MV-RNNs* represent every word and phrase in the syntactic tree with a vector that represents its inherent meaning, and a matrix which describes how the word or phrase alters the meaning of nearby constituents. They evaluated on the SemEval-2010 relation classification dataset (Hendrickx et al., 2010) and report improved performance relative to an SVM and a maximum entropy classifier baseline.

Zeng et al. (2014) proposed a convolutional neural network for relation classification. Their model makes use of a concatenation of lexical and sentence level features extracted from each relation instance. They evaluated their approach on the SemEval-2010 corpus and compared to baseline methods, including the Matrix-Vector Recursive Neural Networks proposed in Socher et al. (2012), and report performance improvements compared to the baselines.

Zhang and Wang (2015) proposed a Recurrent Neural Network (RNN) model for relation classification. The advantage of using RNNs in their approach, as compared to using CNNs, is that they are able to account for long range dependencies in text. They conducted experiments on two datasets, the SemEval-2010 dataset and the KBP37 dataset used by Angeli et al. (2014). They compared to the models of Socher et al. (2012) and Zeng et al. (2014) and report performance improvements compared to both models. Zhang et al. (2017) proposed a dataset (*TACRED*) and an LSTM-based model for relation classification. Their model utilizes an attention mechanism which takes into account the position of the subject and object entities in a sentence for relation classification. They report that their model outperformed various CNN and LSTM baselines on their dataset.

Adel et al. (2016) compared the performance of CNNs to other models such as Support Vector Machines for relation classification. They report that the performance of the models varies per relation class, and that a combination of different models achieves the best performance. Vu et al. (2016) propose to combine both recurrent neural networks and convolutional neural networks for relation classification, in order to benefit from the inductive biases inherent in both models. The authors report that the combination of both models using a simple voting scheme outperformed various baselines on the SemEval-2010 dataset, including the models proposed in Socher et al. (2012), Zeng et al. (2014) and Zhang and Wang (2015).

2.2.6 Relation Classification with Matrix Factorization

A matrix-factorization approach for extracting relations was proposed by Riedel et al. (2013), in the context of *universal schema*, which unifies the relation schemas used by OpenIE extraction methods (surface patterns) and predicates used by structured knowledge bases such as Wikidata and YAGO, into a unified schema. The joint schema makes it possible to easily infer asymmetric implications among relational predicates, such as that `"#A started #B in his garage" => founder_of(A,B)`, and that its converse `founder_of(A,B) => "#A started #B in his garage"` is not necessarily true. Riedel et al. (2013) further model relation detection as a low-rank matrix factorization problem. Within the matrix, observed relational facts are given a value of 1. During training, model F by Riedel et al. (2013) learns latent feature representations for entity pairs and relations with a ranking objective which scores observed relational facts higher than unobserved ones. Prediction for an unobserved (candidate) fact is made my measuring the compatibility between the learned latent representation of the entity pair and relation making up the candidate fact. In evaluations conducted on the NYT dataset (Sandhaus, 2008), the authors show that their approach improves in performance compared to approaches proposed by Mintz et al. (2009); Yao et al. (2011); Surdeanu et al. (2012a) for extracting relations.

2.3 Evaluation Metrics for Relation Classification

A number of evaluation metrics are employed to measure the performance of relation classification systems. This section discusses these metrics , after defining some basic terms.

True Positives: The instances predicted by a classifier as instances of a relation, and which are truly instances of the predicted relation.

True Negatives: The instances predicted by a classifier as not instances of a relation, and which are truly not instances of the relation.

False Positives: The instances predicted by a classifier as instances of a relation, but which are not instances of the predicted relation.

False Negatives: The instances predicted by a classifier as not instances of a relation, but which are instances of the relation.

Accuracy: This is defined as the ratio of correct predictions to that of all predictions, i.e,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.3}$$

This metric can be used for tasks which are balanced in regard to the number of instances of each class.

Precision: This is the ratio of a classifier's correctly predicted instances to all instances predicted for a specific class:

$$Precision = \frac{TP}{TP + FP} \tag{2.4}$$

Recall (or True Positive Rate (TPR)): This is the ratio of a classifier's correctly predicted instances to that of all instances in a specific class:

$$Recall = \frac{TP}{TP + FN} \tag{2.5}$$

False Positive Rate (FPR) (or Fallout): This is the ratio:

$$FPR = \frac{FP}{FP + TN} \tag{2.6}$$

Specificity (or True Negative Rate (TNR)) : This is the ratio of true negatives to all negatives:

$$Specificity = \frac{TN}{TN + FP} = 1 - FPR \tag{2.7}$$

F-Measure (F1) : This is a single score which measures the harmonic mean of precision and recall:

$$F_1 = \frac{2PR}{P + R} \tag{2.8}$$

Im multi-class relation classification, the F_1 score can be computed in one of two ways: Macro F_1 computes the precision, recall and F_1 scores independently for each relation, then takes the average of F_1 scores across relations to obtain the overall F_1 score. On the other hand, Micro F_1 aggregates the true positives, false positives and false negatives of all relations together to compute the final F_1 score.

2.3.1 Ranked Evaluation Measures

In some settings, especially when the output of the extractor is a ranked, e.g. a ranked list of entity pairs that are likely to take part in a relation, evaluation measures inspired by those used to evaluate information retrieval systems are often used. We discuss some of these measures next.

Precision at k: This is the value of precision computed using the top k ranked results.

Average Precision (AP) : Given a ranked list of relation instances, we can compute the precision at every point where a true positive instance is retrieved. The average of the values obtained this way across the entire ranked list is known as average precision.

Mean Average Precision (MAP): This is the mean of the average precision value across all relations of interest.

Precision-Recall Curve This is a plot of precision versus recall at various thresholds. It illustrates how precision changes with recall in the ranked list of results. The area under this curve (Area under the Precision-Recall Curve) can be used to give an aggregate measure of the performance of an extractor which outputs a ranked list of extractions.

Receiver Operating Characteristic (ROC) Curve This is a plot of the true positive rate (recall) versus the false positive rate. It illustrates how recall level changes with false positive rate at various thresholds. The area under this curve (Area under the ROC Curve) can also be used to evaluate the performance of a relation classifier.

2.4 Summary

In this chapter we surveyed relevant background work in relation classification. We described various supervision settings and model choices that can be used to carry out the task, and different metrics that can be employed to evaluate the performance of relation classification systems.

The next chapter discusses our first proposed approach for improving performance of relation classification with limited supervision, based on the use of contextual surface patterns.

Chapter 3

Contextual Pattern Embeddings for Relation Classification

As the survey of relation classification in the last chapter indicates, the task of extracting relations has been widely studied, and a variety of approaches have been proposed for it. A well-known method for extracting relations from a corpus of documents is *universal schema* (Riedel et al., 2013), which is based on matrix factorization. However, in order to perform well on new relations, this approach requires a lot of annotated examples of each new relation for training. In order to address this problem, existing approaches Rocktäschel et al. (2015); Demeester et al. (2016) propose combining matrix factorization with propositional logic rules. Rocktäschel et al. (2015) assumed that there is sufficient training data for new relations that can be used to generate such rules, while Demeester et al. (2016) assumed the availability of relevant external resources such as WordNet (Miller, 1995), from which such rules can be generated. However, such relevant resources are not always available in all domains, and even when available, are not guaranteed to be up-to-date and complete. Thus, they can not always be relied upon as a source of additional supervision for relations with limited supervision instances. An approach that does not assume sufficient initial training data, and does not rely on the use of external resources, but instead effectively exploits supervision signals already present within the corpus, is thus desirable for providing additional supervision to models for extracting new relations.

This chapter investigates such an approach. We propose and show how contextual surface patterns, incorporated within a factorization machines

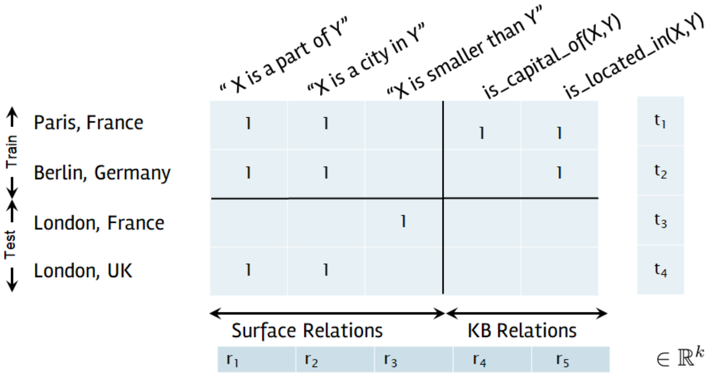


Figure 3.1 The Universal Schema matrix

framework (Rendle, 2010), can be utilized in the same experimental setup as Rocktäschel et al. (2015) and Demeester et al. (2016), for extracting new knowledge base relations without assuming the availability of relevant external resources to provide additional supervision. We test our approach on the standard New York Times dataset (Sandhaus, 2008), and compared to using matrix factorization with propositional logic rules, as proposed by Rocktäschel et al. (2015) and Demeester et al. (2016), and find that with limited training data, our approach obtains gains in performance comparable to the previous state-of-the-art approaches that utilize matrix factorization combined with additional supervision signals in the form of propositional logic rules. In addition, when using the full training data our approach obtains competitive performance compared to matrix factorization-based baselines.

The rest of the chapter is structured as follows. We start with a detailed discussion of extracting relations with universal schema in Section 3.1. In Section 3.2 we discuss the limitations of the previous approaches for extracting relations with limited supervision in the framework of universal schema. Section 3.3 describes our proposed solution to the limitations. It also presents our experimental results and discusses our findings. We describe other relevant work in Section 3.4 and conclude with a summary in Section 3.5.

3.1 Relation Classification with Universal Schema

Universal Schema (Riedel et al., 2013) is an approach to relation classification that jointly embeds textual surface patterns, knowledge base relations and entity pairs in a common embedding space through matrix factorization. It sidesteps the problem of aligning relations to sentences from the training corpus, which can generate noisy training data and cause sub-optimal performance of distantly supervised relation classification approaches. It achieves this by performing joint inference across surface patterns, knowledge base relations and entity pairs. This approach casts the problem of extracting relations between entities as one of link prediction over a universal schema consisting of the union of textual surface patterns, structured knowledge base relations and entity pairs. The universal schema matrix is as illustrated in Figure 3.1, with entity pairs in the rows and relations in the columns. Each cell of the matrix represents a relation between two entities, where cells with value 1 are observed relations and the empty ones are unobserved relations we would like to predict. The framework learns latent feature vectors for relations and entity pairs by factorizing this matrix, and utilizes facts from text and the knowledge base stipulating that a certain relation holds among two entities to provide supervision signals for relation classification.

Formally, let $\mathcal{T}$ and $\mathcal{R}$ be the set of entity pairs and relations (structured knowledge base and surface relations) respectively. Each entity pair $t = (e_1, e_2)$ consists of the subject e_1 and the object e_2 entities respectively. Given a matrix $\boldsymbol{M}$ (for instance Figure 3.1), consisting of observed positive facts (cells with value 1) and unobserved facts (the empty cells), we would like to be able to complete the matrix by predicting the missing values. One well-established way to achieve this is by factorizing the matrix M. This can be conceived as a matrix factorization problem of the form:

$$\boldsymbol{M} = \boldsymbol{T}\boldsymbol{R}^\top \quad \in \mathbb{R}^{|\mathcal{T}|\times|\mathcal{R}|} \tag{3.1}$$

where $\boldsymbol{T} \in \mathbb{R}^{|\mathcal{T}|\times k}$ and $\boldsymbol{R} \in \mathbb{R}^{|\mathcal{R}|\times k}$ are the matrices containing latent vector representations of entity pairs and relations respectively.

Model **F** by Riedel et al. (2013) represents each entity pair $t \in \mathcal{T}$ and relation $r \in \mathcal{R}$ by latent vectors $\boldsymbol{\theta}_t \in \mathbb{R}^k$ and $\boldsymbol{\theta}_r \in \mathbb{R}^k$ respectively, and models the score s of a fact $f = r(e_1, e_2)$ as:

$$s(f) = \langle \boldsymbol{\theta}_t, \boldsymbol{\theta}_r \rangle \tag{3.2}$$

where $\langle ., . \rangle$ is the scalar (dot) product.

The score is afterwards mapped to a probability with the logistic function:

$$p(f) = \sigma\Big(s(f)\Big) = \frac{1}{1 + \exp\Big(-s(f)\Big)} \tag{3.3}$$

Since each relation and entity pair is represented by a low-dimensional vector representation, this leads to similar relations (and entity pairs) having representations that are close in latent space.

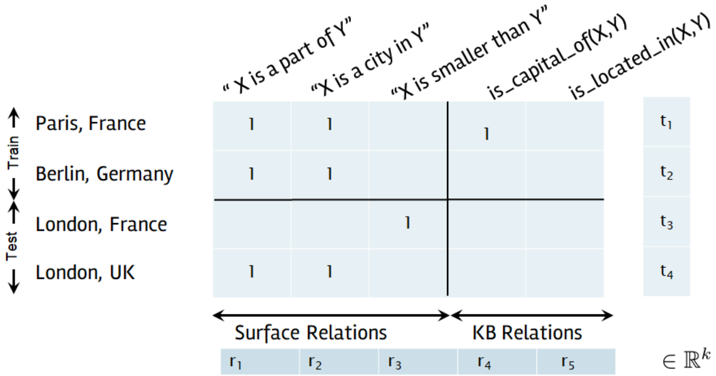
3.1.1 Learning

An important consideration for learning in the universal schema framework is that the matrix consists of only positive data, i.e., the observed facts, and does not have any explicit negative data since negative facts are not observed. Learning from positive-only data is known as *implicit feedback* (Rendle et al., 2009a) in the recommendation literature, where the task is to recommend for users interesting items given their history of known item interactions, e.g purchases or ratings. In order to address this problem, Rendle et al. (2009a) proposed Bayesian Personalized Ranking (BPR) which learns parameters that ranks observed user interactions higher than unobserved ones. The assumption underlying BPR is that unobserved user interactions are not necessarily negative interactions, but should be given lower scores than observed positive interactions by the learned model.

In the framework of universal schema, one can think of a relation as the user for which we want to recommend entity pairs given the other entity pairs which are known to be participating in that relation. Thus, Riedel et al. (2013) also adopt a BPR ranking objective to estimate model parameters. Specifically, given an observed tuple (e_1, e_2) of a relation r during training, negative training data is generated for r by sampling other entity pairs (e_3, e_4) which have not been observed to be participating in the relation. Parameters are then estimated such that the model gives an observed positive fact $f^+ = r(e_1, e_2)$ a higher score than a sampled unobserved fact $f^- = r(e_3, e_4)$. The objective maximized by the model can then be expressed as:

$$\mathcal{L} = \sum_{\substack{f^+ \in \mathcal{F}^+ \\ f^- \in \mathcal{F}^-}} \log \left(\sigma \left(s(f^+) - s(f^-) \right) \right) \tag{3.4}$$

where $\mathcal{F}^+$ is the set of all observed facts and $\mathcal{F}^-$ is the set of all sampled negative facts.



	" X is a part of Y"	"X is a city in Y"	"X is smaller than Y"	is_capital_of(X,Y)	is_located_in(X,Y)	
Paris, France	1	1		1		t_1
Berlin, Germany	1	1				t_2
London, France			1			t_3
London, UK	1	1				t_4
	r_1	r_2	r_3	r_4	r_5	

Figure 3.2 The Universal Schema matrix without training instances of the relation `is_located_in(X,Y)`

3.2 Predicting Relations with Limited Supervision

A major weakness of approaches like universal schema is that they are unable to accurately predict knowledge base relations whose instances were unobserved at training time. This problem is illustrated in Figure 3.2, where no training instances of the relation `is_located_in(X,Y)` are available for training. At test time, any predictions for this relation would essentially be random predictions, since the model would not have been able to learn any useful representations for the relation.

To address this problem, Rocktäschel et al. (2015) and Demeester et al. (2016) propose to inject prior knowledge in the form of propositional logic rules to improve performance for relations with zero or few training labels. Such rules, which can be of the form `X is a city in Y => is_located_in (X,Y)`, can for instance be obtained from domain experts or mined from a resource such as WordNet (Miller, 1995). They can then be injected into the universal schema matrix prior to, during, or after matrix factorization, to enable the extraction of relations with limited training instances. As an illustration, injecting the rule `X is a city in Y => is_located_in (X,Y)` before matrix factorization turns Figure 3.2 into Figure 3.1. This process essentially adds two training instances for the relation `is_located_in (X,Y)` into the matrix in Figure 3.2, which can afterwards be factorized to extract relations.

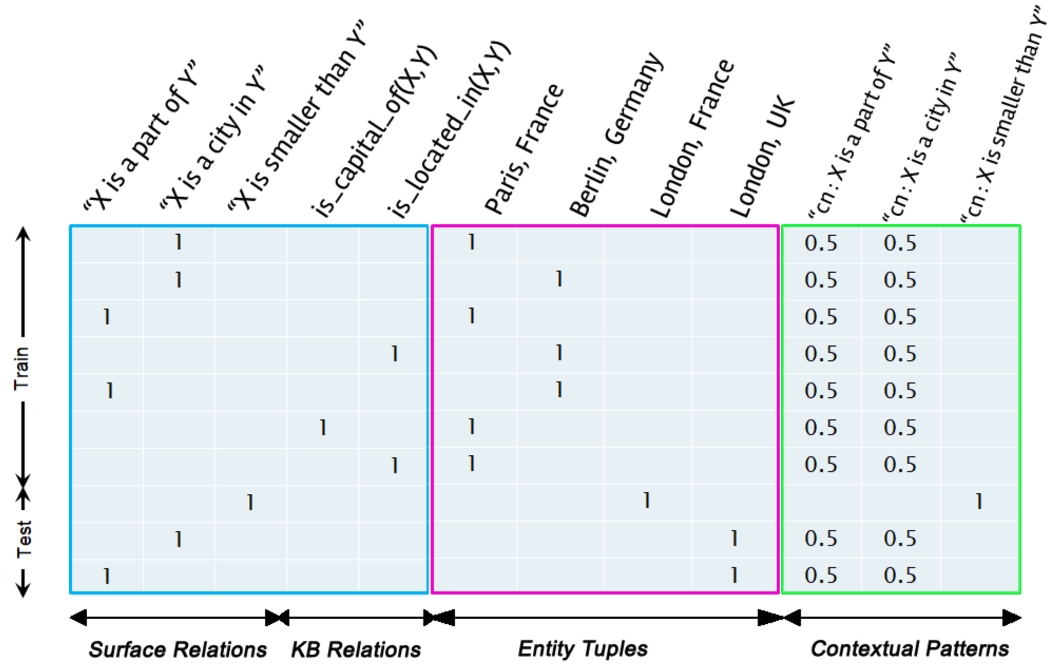
In order to utilize this approach, we need to first obtain or generate such rules. Rocktäschel et al. (2015) generated such rules from the training data, under the assumption that new relations have sufficient training data that can be used for this purpose. However, this is not always the case, as it is possible to have new relations for which we have very few, or even a single training instance in the universal schema matrix. In such cases, the approach proposed by Rocktäschel et al. (2015) may not be effective, since the method relies on an initial training set for rule extraction. Another option for generating such rules was proposed by Demeester et al. (2016), who assumed the availability of an external resource such as WordNet, from which such rules can be generated. The weakness of this approach is that such external resources are not guaranteed to be available for all domains.

As an alternative solution to address these problems, we explore the use of contextual surface patterns which are already present in many textual corpora to provide additional supervision for extracting new relations. Intuitively, the occurrence of contextual patterns like `X is a city in Y` and `X is a part of Y` with an entity pair should make such an entity pair more likely to be an instance of the `is_located_in (X,Y)` relation. Even though the occurrence of such patterns may be sparse in a text corpus, our hypothesis is that explicitly modelling them and their interactions can serve as a good source of additional supervision for extracting knowledge base relations with limited training instances. The next section describes our approach to modelling such patterns and their interactions.

3.3 Factorization Machines for Relation Classification with Limited Supervision

This section describes our approach for extracting relations with limited supervision, which investigates the usefulness of modelling contextual surface patterns and their interactions in computing the probability score of facts when supervision is limited. We start with a description of Factorization Machines (FMs) (Rendle, 2010, 2012) on which we develop our approach.

Factorization Machines (FM) are a generalization of matrix factorization proposed in the context of recommender systems as a way to learn effective scoring functions with sparse inputs, in order to assess how likely it is that a user-item combination occurs in reality. More concretely, they model the scoring of a possibly sparse, real-valued input feature vector $\mathbf{f} \in \mathbb{R}^d$ according to the following equation:



	Surface Relations			KB Relations		Entity Tuples				Contextual Patterns		
	“X is a part of Y”	“X is a city in Y”	“X is smaller than Y”	is_capital_of(X,Y)	is_located_in(X,Y)	Paris, France	Berlin, Germany	London, France	London, UK	“cn: X is a part of Y”	“cn: X is a city in Y”	“cn: X is smaller than Y”
Train		1				1				0.5	0.5	
		1					1			0.5	0.5	
	1					1				0.5	0.5	
					1		1			0.5	0.5	
	1						1			0.5	0.5	
				1		1				0.5	0.5	
					1	1				0.5	0.5	
Test			1					1				1
		1							1	0.5	0.5	
	1								1	0.5	0.5	

Figure 3.3 Input observations as a matrix with contextual pattern information. Each row in the matrix represents a fact (an instance), and the three variables with names starting with “cn:” are contextual patterns.

$$s(\mathbf{f}) = \sum_{m=1}^{d} b_m f_m + \sum_{m=1}^{d} \sum_{n=m+1}^{d} \langle \phi_m, \phi_n \rangle f_m f_n \tag{3.5}$$

The first summand is a linear model, where each feature f_m is weighted by a corresponding feature weight $b_m \in \mathbb{R}$. The second summand captures the interaction between all possible feature pairs under a low-rank assumption. Each feature f_m has a corresponding embedding $\phi_m \in \mathbb{R}^k$ with $k << d$, and the interaction between two features is captured via their dot product $\langle \phi_m, \phi_n \rangle$ multiplied by the product of their values in the instance $f_m f_n$. The dot products among all feature pairs represent the weights that we would have in a model having a weight for each feature combination ($d(d-1)/2$ weights) but with fewer parameters ($d(1+k)$) and thus easier to learn from less and/or sparse data. Modelling feature interactions in the context of recommender systems, since some features would represent the item and others the user, and the linear component of the model would only capture that some users tend to buy more items or that some items are more popular among users. The feature embeddings that are used to capture their interactions can inform us whether a particular user will buy a particular item. The above equation represents an order-2 FM which captures interactions between pairs of features, but higher order FMs can capture interactions among feature groups of higher cardinality at additional computational cost. In this work, we make use of order-2 FMs due to their linear time computational cost with respect to the input size (Rendle, 2010).

An alternative view is that we learn the rank-k factorization of the matrix containing the weights for each feature pair, hence the name factorization machines. Rendle (2010) showed that a FM model is effective in several learning settings, even those with sparse features, and is also capable of approximating the behaviour of many matrix and tensor factorization models.

A FM model learns factorized interaction parameters for all its features and their interactions, which can be very helpful since surface pattern features are sparse and supervision for new relations can be limited. We leverage this ability to learn feature interactions from sparse data to incorporate contextual information into our relation classification approach to improve its performance with limited data.

In order to apply FMs for learning relation extractors, we represent a candidate fact as a triple (r, t, c^t) consisting of a relation $r \in \mathcal{R}$, an entity pair $t \in \mathcal{T}$ and the contextual surface patterns c^t of the entity pair. The contextual surface patterns represent the counts of surface patterns that have been observed together with tuple t in a text corpus, normalized to sum to one. We generate $\mathbf{f}$, the fact feature vector by concatenating vectors encoding each of these elements. The relation r and tuple t are encoded as one-hot feature vectors of dimensionalities $|\mathcal{R}|$ and $|\mathcal{T}|$ respectively. Thus, relations, entity tuples and contextual patterns constitute the variables of our model, and are the only variables we learn embeddings for with our approach. [1]

The intuition behind using the contextual surface pattern features being that they provide evidence of the surface patterns that are descriptive of the entity pair in the text corpus, allowing the model to learn which combinations of surface patterns are indicative of which knowledge base relations. Hence, the model is able to draw on statistical evidence from surface patterns across a text corpus in order to derive more reliable estimates for the interaction factors of relations. This also gives us the benefit of making the most of surface relations, which are easily obtained but noisy, to learn with very few annotation labels for new relations.

[1] We did not learn embeddings for the other possible variables such as the individual entities in a tuple or their types.

In the left-hand part of the matrix in Figure 3.3 each instance has either a surface relation or a KB relation active, thus their correlations may be ignored unless we consider the contextual patterns on the right-hand side. For example, the first row in Figure 3.3 represents that the tuple `Paris,France` was observed with the surface relation `X is a city in Y` and that the same tuple was observed with two contextual surface patterns, `cn:X is a city in Y` and `cn:X is a part of Y`, hence each of them have a value of 0.5. Similarly, the sixth row represents that the same tuple `Paris,France` with the same contextual patterns having the KB relation `is_capital_of(X,Y)`. This allows the FM model to learn the interaction between the surface patterns `X is a city in Y` and `X is a part of Y` and the KB relation `is_capital_of(X,Y)`. Furthermore consider that we want to predict which is a more likely entity tuple between `London,UK` and `London,France` for the knowledge base relation `is_located_in(X,Y)`. Observe that the tuples `London, UK` and `Paris, France` have more contextual surface pattern overlap than the tuple `London,France`. The proposed model would be aware of such correlations to give a higher score for the fact `(London,is_located_in,UK)` than `(London,is_located_in,France)`. We henceforth refer to the Factorization Machines model without contextual pattern embeddings as *FM* and that with contextual pattern embeddings as *FMC*.

3.3.1 Objective Formulation

Given a text corpus, we aim to extract relations between entities with limited training data for each relation by learning a model that can differentiate between true and false facts, i.e. assign high scores to the former and lower scores to the latter using Equation 3.5. However, only examples of observed true relations between entities (positive facts) are available at training time. In order for the model to effectively discriminate between positive and negative facts, it needs to have also seen examples of negative facts. One way to achieve this is to treat observed relations as true facts and all unobserved relations between entities as false facts. However since the facts we seek to extract are unobserved, this carries the risk that we treat plausible relations between entities as negative, which can consequently lead to inferior model performance. Following the work of Riedel et al. (2013), we make use of the alternative approach of treating unobserved facts as unknowns, and left for the model to infer. This is achieved using a ranking-based objective, which optimizes to rank observed facts higher that unobserved ones. Concretely, we make use of the Bayesian Personalized Ranking (BPR) (Rendle et al., 2009b) objective, which optimizes for the maximal difference between the score of observed and unobserved facts. Given a set of observed $\mathcal{F}^+$ and unobserved $\mathcal{F}^-$ facts, we estimate model parameters Θ that satisfy the following objective:

$$\min_{\Theta} -\sum_{\substack{f^+ \in \mathcal{F}^+ \\ f^- \in \mathcal{F}^-}} \log\left(1 + e^{\delta(f^+, f^-)}\right) + \lambda \|\Theta\|_2^2 \tag{3.6}$$

where $\delta(f^+, f^-) = s(f^+) - s(f^-)$ and λ is a regularization hyper parameter. The objective in Equation 3.6 essentially maximizes the difference $\delta(f^+, f^-)$ between the scores of observed and unobserved facts.

Since the set $\mathcal{F}^-$ is unobserved, it is generated automatically from $\mathcal{F}^+$ by random sampling. Specifically, in each iteration and for every positive fact f^+ in the current batch, we fix the relation r and randomly select an entity pair $t' \in \mathcal{T}$, such that $(r, t', c^{t'})$ has not been observed.

3.3.2 Dataset

For our experiments, we make use of the dataset of Riedel et al. (2013), which consist of data from New York Times (NYT) corpus (Sandhaus, 2008). The corpus has been preprocessed with a named entity recogniser and the entities have been linked, where possible, with their corresponding Freebase (Bollacker et al., 2008b) entities. The shortest dependency path between each pair of entities in a sentence has also been extracted as the surface relation.

3.3.3 Setup and Evaluation

We use the same dimensionality for the embeddings and the same preprocessing (named entity recognition and linking, syntactic parsing) as the other approaches we compare with in order to ensure a fair comparison. For all experiments, we make use of a latent dimension size of 100, L_2 regularization penalty of 0.01, and ran our model for 1000 epochs. Our approach is implemented in Tensorflow (Abadi et al., 2016), and uses Adam (Kingma and Ba, 2014) for optimization, with a learning rate of 1×10^{-4} and batch size of 1024. We sample one unobserved fact at random per positive fact during training.

We make use of the same evaluation setup as Riedel et al. (2013), who retrieved for each relation the top 1000 entity tuples from each system, the top 100 of which is then pooled and manually annotated. These provided a set of results that is used to compute precision measures for each system. We computed Mean Average Precision (MAP) and weighted Mean Average Precision (wMAP) for each run. While MAP computes the mean of average precision scores across all the relations for each system, weighted MAP takes into account the number of true facts for each relation.

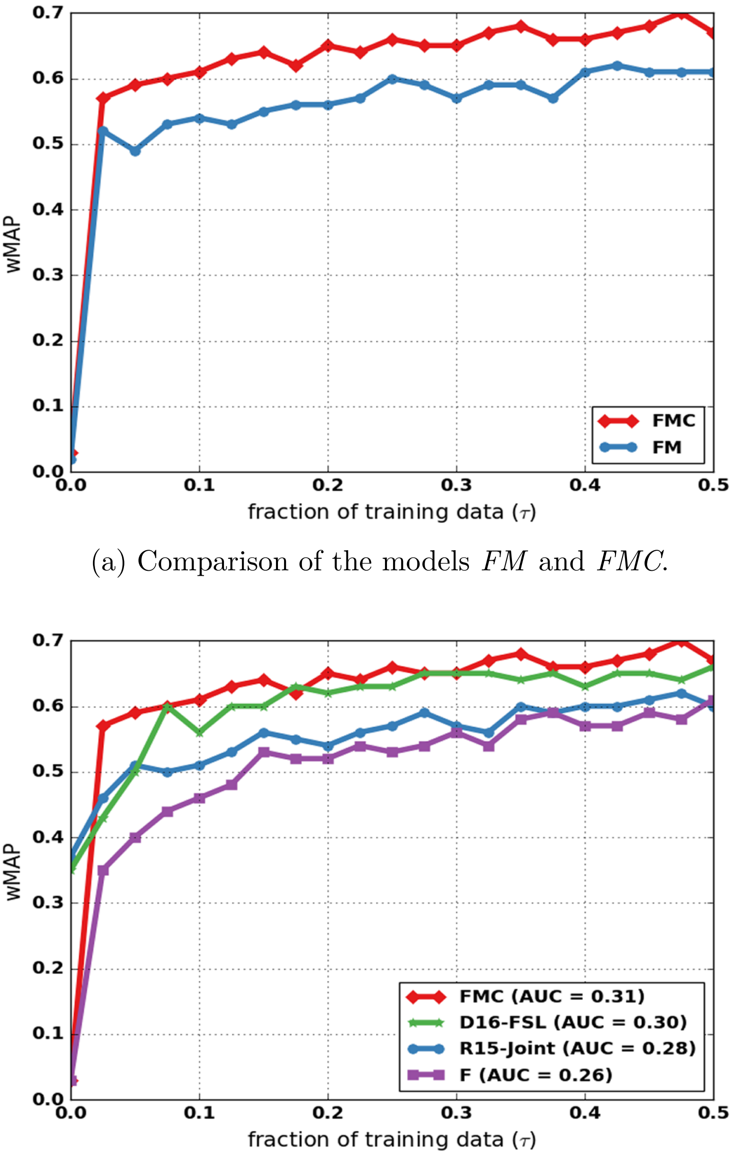
Relation	#	M09	Y11	S12	R13-N	R13-F	R13-NF	R13-NFE	FMC
person/company	104	0.67	0.63	0.69	0.72	0.75	0.75	0.78	**0.80**
location/containedby	75	0.48	0.51	0.53	0.42	**0.68**	0.66	**0.68**	**0.68**
person/nationality	30	0.13	**0.38**	0.12	0.13	0.18	0.18	0.20	0.20
author/works_written	29	0.50	0.51	0.52	0.45	0.61	0.63	**0.69**	0.67
parent/child	19	0.14	0.25	0.62	0.46	0.76	0.78	0.76	**0.79**
person/place_of_death	19	0.79	0.79	0.86	**0.89**	0.83	0.85	0.86	0.83
person/place_of_birth	18	0.78	0.75	0.82	0.50	0.83	0.81	**0.89**	0.81
neighborhood/neighborhood_of	12	0.00	0.00	0.08	0.43	0.65	0.66	**0.72**	0.62
person/parents	7	0.24	0.27	**0.58**	0.56	0.53	**0.58**	0.39	0.56
company/founders	4	0.25	0.25	0.53	0.24	0.77	**0.80**	0.68	0.67
film/directed_by	4	0.06	0.15	0.25	0.09	0.26	0.26	**0.30**	0.07
sports_team/league	4	0.00	0.43	0.18	0.21	0.59	**0.70**	0.63	0.48
team/arena_stadium	3	0.00	0.06	0.06	0.03	0.08	**0.09**	0.08	**0.09**
team_owner/teams_owned	2	0.00	0.50	0.70	0.55	0.38	0.61	**0.75**	0.63
roadcast/area_served	2	**1.00**	0.50	**1.00**	0.58	0.58	0.83	**1.00**	0.58
structure/architect	2	0.00	0.00	**1.00**	0.27	**1.00**	**1.00**	**1.00**	**1.00**
composer/compositions	2	0.00	0.00	0.00	0.50	0.67	**0.83**	0.12	**0.83**
person/religion	1	0.00	**1.00**	**1.00**	0.50	**1.00**	**1.00**	**1.00**	**1.00**
film/produced_by	1	**1.00**	**1.00**	**1.00**	**1.00**	0.50	0.50	0.33	**1.00**
MAP		0.32	0.42	0.55	0.45	0.61	0.66	0.63	0.65
Weighted MAP		0.48	0.51	0.56	0.52	0.66	0.66	0.68	0.68

Table 3.1 Results using the full training dataset. The # column is the number of true facts in the test pool.

3.3.4 Results and Discussion

Limited Supervision In the limited supervision experiments, we perform evaluations with a fraction $\tau \in [0, 0.5]$ of the training labels for each relation. Figure 3.4a presents the results of limited supervision experiments for the two variants of our approach FM and FMC. The figure shows that the difference in performance between models FM and FMC is wider when less supervision data is available. These results demonstrate that the contextual information incorporated by model FMC enhanced its performance when less supervision labels are available to the model.

Figure 3.4b presents results of FMC compared to state-of-the-art models from Rocktäschel et al. (2015) (R15-Joint) and Demeester et al. (2016) (D16-FSL). Note though that this comparison is not fair to our approach, since it does not make use of any rules to generate extra supervision data, and this affected its performance when there are zero supervision instances (when $\tau =$



(a) Comparison of the models *FM* and *FMC*.

(b) Comparison of model *FMC* with previous work. Results obtained from (Demeester et al., 2016).

Figure 3.4 Comparison of model *FM* with *FMC* (a), and *FMC* with previous work (b).

Relation	# patterns
location/containedby	786
person/company	332
person/nationality	235
author/works_written	229
person/place_of_birth	216
person/place_of_death	117
parent/child	77
neighborhood/neighborhood_of	74
film/directed_by	25
company/founders	25
sports_team/arena_stadium	23
sports/teams_owned	19
person/religion	16
film/film/produced_by	15
person/parents	12
sports_team/league	9
broadcast/area_served	7
structure/architect	7
composer/compositions	6

Table 3.2 Number of associated surface pattern fact mentions of each relation in the training set.

0.0). Nevertheless, it was still able to obtain better coverage, as measured by the wMAP AUC.

Full Supervision We also perform experiments on the full training portion of the dataset to investigate the model's performance when full supervision is available. Table 3.1 presents the results for several approaches that do not use external supervision data from the literature (M09: Mintz et al. (2009), Y11: Yao et al. (2011), S12: Surdeanu et al. (2012b), R13-*: Riedel et al. (2013)), and our FM implementation with contextual information (FMC). When full supervision is available, FMC is still able to obtain competitive performance with the use of contextual information. FMC performs well on both relations with very few true facts in the test pool and those with the most facts in the test pool.

Table 3.2 presents the relations with the most surface patterns in the training corpus. The relations that model FMC performs well on in Table 3.1 tend to rank high on this table, e.g. *location/contained_by* and *person/company*, which suggests that the incorporation of contextual surface patterns within a FM model is useful for better modelling of knowledge base relations in general.

3.4 Related Work

Welbl et al. (2016) and Petroni et al. (2015) also explored the use of factorization machines for extracting relations. Welbl et al. (2016) explored the use of *bigrams*, which are pairs selected from $\{e_1, r, e_2\}$, as the variables of their model, while Petroni et al. (2015) proposed using additional features such as document metadata. However, such features are not always available. We propose using contextual surface patterns, since these are always available in the data, which gives our approach the ability to capture correlations between surface patterns and knowledge base relations that is not possible in both of these approaches. In addition, neither Welbl et al. (2016) nor Petroni et al. (2015) investigated the effectiveness of their approach for extracting relations in the limited supervision setting.

3.5 Summary

This chapter considered the task of learning to extract relations with limited supervision data. We proposed a FM based model that utilized contextual surface patterns, which are readily available within the data, as additional supervision when labelled data for relations is limited. We showed that our approach improved extraction performance compared to previous approaches and obtained competitive results when full supervision is available.

The approach we have presented in this chapter needs at least one instance of each relation during training in order to be able to make predictions for the relation at test time. In the next chapter we investigate an approach that is able to make predictions for new relations at test time.

Chapter 4

Zero-shot Relation Classification via Textual Entailment

The previous chapter proposed an approach for extracting relations with limited supervision based on the use of a Factorization Machines model. The major weakness of this approach is that it assumes the availability of at least one supervision instance for each relation. This implies that the approach is not applicable for relation classification when there are zero supervision instances for some relations. This chapter proposes an approach for relation classification without any annotated instances, by framing the task as that of textual entailment.

This reformulation brings a number of advantages. First, we are able to utilize relation descriptions to provide supervision for classifying relations which have no labelled instances. Relation descriptions are easy to obtain, and are also available as part of many relation ontologies, making them a readily available source of supervision. The second advantage of our approach is that we are able to leverage existing textual entailment resources, such as datasets and models, for relation classification. For instance, we can pre-train a textual entailment model on a textual entailment dataset, and use the pre-trained model for zero-shot relation classification. Finally, our approach allows us to seamlessly combine any available supervision data for relations together with data from textual entailment to provide additional supervision for relation classification. In our experiments, we demonstrate that this combination leads to improved performance.

The rest of this chapter is organized as follows. Section 4.1 starts by providing the background on textual entailment needed to understand our approach. We define textual entailment and discuss a number of tasks, datasets and models that have been proposed for it. We next describe our approach of relation classification by textual entailment in Section 4.2, where we find that in contrast to previous relation classification models, our approach is able to perform zero-shot classification of relations without training instances. We discuss related work in Section 4.3 and thereafter conclude with a summary in the last section.

4.1 Background on Textual Entailment

The task of recognising textual entailment (also referred to as natural language inference) considers the directional relationship between two fragments of text. It evaluates whether the meaning of a fragment of text (the hypothesis) can be inferred given another fragment of text (the premise). If the meaning of the hypothesis follows from that of the premise, then the premise entails the hypothesis and an *entailment* relation holds between the two text fragments. A *contradiction* relation exists in the case when the meaning of the hypothesis contradicts that of the premise, and a *neutral* relation is said to exist when the pair of text neither contradict nor entail each other.

The Recognising Textual Entailment tasks (Dagan et al., 2005) were proposed as a way to enable the development of textual entailment models that are transferable to other natural language tasks such as information extraction and question answering, amongst others. Since the introduction of this task, various textual entailment models and approaches have been developed for it (Bos and Markert, 2005; Jijkoun and de Rijke, 2005; Roth et al., 2009; Glickman et al., 2006). These approaches are based on the use of shallow lexical features, and rely on surface form similarity for their predictions.

To enable the development of approaches which require the availability of reasonably large quantities of supervision data, Bowman et al. (2015) released the Stanford Natural Language Inference (SNLI) corpus and reported the performance of a number of baseline approaches on the corpus. The release of this dataset accelerated the development of neural models and approaches for textual entailment. For instance, Rocktäschel et al. (2016); Bowman et al. (2016); Parikh et al. (2016); Chen et al. (2016a) have proposed various neural network-based models based on LSTMs (Hochreiter and Schmidhuber, 1997). These models were evaluated on the SNLI corpus, and surpassed the performance of the previously introduced models that made extensive use of hand-engineered natural language pipelines. Recently, a multi-genre version of this dataset was released by Williams et al. (2017).

A simple baseline approach, when learning representations to determine entailment between pairs of text, is to treat the pair of input as separate fragments of text and independently learn a representation for each fragment. We can then concatenate the learned representations to make predictions. However, a model that works this way may not be able to fully account for any intra- and/or inter-dependencies in its inputs to reason effectively about the entailment of word and phrase pairs in the premise and hypothesis. The model proposed by Rocktäschel et al. (2016) made use of conditional encoding and an attention mechanism to address this problem. Instead of treating the premise and hypothesis as independent pieces of text, conditional encoding learns a hypothesis representation that is dependent (that is, conditioned) on the representation learned for the premise. They further introduced a *word-by-word* attention mechanism, which they found allowed their model to reason explicitly about pairs of words and phrases in the premise and hypothesis.

A model that uses tree-structured recursive neural networks to compute representations for the premise and hypothesis was proposed by Bowman et al. (2016). The model works by concatenating its internal representations of both the premise and the hypothesis, their difference, and element-wise product and feeding the result into a number of neural network layers and a linear transformation layer. The output of this process is then finally passed through a softmax layer to make predictions. The approach though is not parameter efficient, as it requires a large number of parameters to obtain modest performance on the SNLI corpus.

Parikh et al. (2016) proposed a decomposable attention model for the task of textual entailment. The model incorporates a novel attention computation step that attends to the words and phrases in the premise and hypothesis in a joint manner. The model also utilizes word position information in the form of an intra-sentence attention mechanism that encodes the compositional relationship between the words in a sentence. When compared to that of Bowman et al. (2016), this model requires fewer parameters and achieves higher accuracy on the SNLI dataset. Chen et al. (2016b) present a hybrid neural model that consists of two sub-models. While one sub-model exploits sequential language information through the use of LSTMs, the other utilizes recursive language information through the use of tree-structured LSTMs. The sequential model replaces the intra-attention mechanism of Parikh et al. (2016) with attentional information derived from bidirectional LSTMs. They show that this change results in gains in the prediction accuracy of their model.

4.2 Relation Classification via Textual Entailment

Given a unit of text containing two entities (subject and object entities), relation classification seeks to determine the relation between the two entities as expressed by the given unit of text. In order for humans to be able to determine if a relation is expressed in a text fragment, they make reference to some predefined notion or meaning of the specific relation. This meaning can be obtained from prior knowledge, or from a lexicon of definitions. This information is also available as part of many relation ontologies, for instance Freebase or Wikidata, in the form of relation descriptions. For instance , the TAC-KBP [1] tasks provide guidelines for task annotators to use in extracting example instances of relations from given document collections, and for human assessors to use in evaluating the correctness of various system outputs. These guidelines include the text descriptions of the relations in their ontology. The assessors use these descriptions as a guide to assess the correctness of the instances of relations extracted by the various systems participating in the task. This indicates that relation descriptions provide useful information in extracting relations in that ontology from document collections. Thus, our hypothesis is that such descriptions should also provide useful information for classifying relations without labelled supervision instances.

If we think of relation descriptions as providing a definition of the meaning of a relation, then this implies we can consider the task of relation classification as one of determining if the meaning of a relation can be inferred between two entities in a given fragment of text. In other words, we can consider the descriptions as the hypothesis of a textual entailment task, with the text fragments as premise and the task of relation classification becomes that of

[1] https://tac.nist.gov/2015/KBP

Relation	Subject (**X**)	Object (**Y**)	Text (Premise)	Description (Hypothesis)
religious_order	Lorenzo Ricci	Society of Jesus	**X** (August 1, 1703 – November 24, 1775) was an Italian Jesuit, elected the 18th Superior General of the **Y**.	***X*** *was a member of the group* ***Y***
director	Kispus	Erik Balling	**X** is a 1956 Danish romantic comedy written and directed by **Y**.	*The director of* ***X*** *is* ***Y***
designer	Red Baron II	Dynamix	**X** is a computer game for the PC, developed by **Y** and published by Sierra Entertainment.	***Y*** *is the designer of* ***X***

Table 4.1 Examples of relations, entities, sample text instances, and relation descriptions.

determining if the meaning of the relation can be inferred from that of the text. Given text from which we wish to determine the presence of certain relations, we can join the description of the relation of interest to each text instance to form a textual entailment instance.

More formally, we propose to formulate the task of relation classification as that of textual entailment as follows. Given a unit of text T which mentions a subject X and a candidate object Y of a knowledge base relation $R(X, Y)$, and a natural language description $d(X, Y)$ of R, we wish to evaluate whether T expresses an instance of $R(X, Y)$. This is equivalent to a textual entailment problem in which the unit of text and the relation description can be considered as the premise P and hypothesis H respectively. For instance, given the relation *spouse_of(X, Y)* and its description *"Y is the spouse of X"* and the text *"**Michelle** is married to **Barack**"*, we formulate this task as one of determining the truthfulness of the hypothesis "**Michelle** is the spouse of **Barack**", given the text. The challenge then becomes that of determining the truthfulness of the hypothesis given the premise. Table 4.1 gives further examples of knowledge base relations and their natural language descriptions.

This reformulation brings a number of advantages. The first is that, just as humans are able to identify instances of relations when given the textual description of a relation, it provides a way to utilize relation descriptions for zero-shot relation classification. We achieve this by pairing to each potential relation instance a description of the candidate relation, as shown in Table 4.1, and training a textual entailment model on the resulting data. At test time, given the description of a new relation and its candidate instances, we can pair the description with the instances to generate premise and hypothesis pairs for classification.

The second advantage of our approach is that it allows us to use existing textual entailment resources, such as datasets and models, for relation classification. As discussed in section 4.1, a number of large, manually annotated datasets and elaborate models have already been proposed for textual entailment, and all of this can be directly applied to the task of relation classification. In the experiments section, we show that we can perform zero-shot relation classification just by pretraining a textual entailment model on an existing textual entailment dataset.

Additionally, our approach allows for the augmentation of any available relation classification data with existing textual entailment data to enhance performance for relation classification. This can be useful, for instance, when there is some supervision data available for relation classification. In this case, we can simply generate a textual entailment dataset from the existing labelled instances by pairing each relation instance with the description of its relation label, and adding this to an existing textual entailment dataset. In our experiments, we show that this approach can lead to improvements in performance for relation classification.

4.2.1 Model

The problem of determining whether the meaning of a piece of text is entailed by another can be handled with a textual entailment model, and we take as our base model the textual entailment model introduced by Chen et al. (2016b). We make use of the Enhanced Sequential Inference Model (*ESIM*) variant, which we briefly described previously. We now present a more detailed description of this model.

ESIM utilizes Bidirectional Long Short-Term Memory (Hochreiter and Schmidhuber, 1997; Graves and Schmidhuber, 2005) (BiLSTM) units as a building block and accepts two sequences of text as input. It then passes the two sequences through three modelling stages - input encoding, local inference modelling and inference composition, and finally a prediction stage which returns the class with the highest classification score, out of the three classes *entailment*, *contradiction* and *neutral*.

In this section we first briefly describe the input encoding, local inference modelling and inference composition stages. We then describe how we adapt the input encoding and inference composition stages using conditional encoding in the following subsection.

The input to the *input encoding* stage is two sequences of vectors, $\{\boldsymbol{p}_i\}$ and $\{\boldsymbol{h}_j\}$, or more compactly two matrices $\boldsymbol{P} \in \mathbb{R}^{I \times d}$ for the premise and $\boldsymbol{H} \in \mathbb{R}^{J \times d}$ for the hypothesis, where I and J are respectively the number of words in the premise and hypothesis, and d is the dimensionality of each vector representation. Then the input sequences are processed with BiLSTM units to yield new sequences $\bar{\boldsymbol{P}} \in \mathbb{R}^{I \times 2d}$ for the premise and $\bar{\boldsymbol{H}} \in \mathbb{R}^{J \times 2d}$ for the hypothesis:

$$\bar{\boldsymbol{P}}\ ,\ \overrightarrow{\boldsymbol{c}}_p, \overleftarrow{\boldsymbol{c}}_p = BiLSTM(\boldsymbol{P}) \tag{4.1}$$

$$\bar{\boldsymbol{H}}\ ,\ \overrightarrow{\boldsymbol{c}}_h, \overleftarrow{\boldsymbol{c}}_h = BiLSTM(\boldsymbol{H}) \tag{4.2}$$

where $\overrightarrow{\boldsymbol{c}}_p\,, \overleftarrow{\boldsymbol{c}}_p \in \mathbb{R}^d$ are respectively the last memory cell states in the forward and reverse directions of the BiLSTM that reads the premise. $\overrightarrow{\boldsymbol{c}}_h\,, \overleftarrow{\boldsymbol{c}}_h \in \mathbb{R}^d$ are similarly defined for the hypothesis.

In the *local inference modelling* stage, soft alignments between words in the premise and hypothesis are computed using neural attention. This stage takes as input the word representations $\bar{\boldsymbol{p}}_i$ and $\bar{\boldsymbol{h}}_j$ obtained from the input encoding stage. It first computes the interaction e_{ij} between each word in the premise and hypothesis as the dot product of their vectors, $e_{ij} = \bar{\boldsymbol{p}}_i^T \bar{\boldsymbol{h}}_j$, and then uses this to compute their soft alignment:

$$\tilde{\boldsymbol{p}}_i = \sum_{j=1}^{J} \frac{\exp{(e_{ij})}}{\sum_{f=1}^{J} \exp{(e_{if})}} \bar{\boldsymbol{h}}_j,\ \forall i = 1, 2, ..., I \tag{4.3}$$

$$\tilde{\boldsymbol{h}}_j = \sum_{i=1}^{I} \frac{\exp{(e_{ij})}}{\sum_{f=1}^{I} \exp{(e_{fj})}} \bar{\boldsymbol{p}}_i,\ \forall j = 1, 2, ..., J \tag{4.4}$$

The stage then computes new vector representations $\boldsymbol{s}_i$ and $\boldsymbol{t}_j$ for each word in the premise and hypothesis as:

$$\boldsymbol{s}_i = [\bar{\boldsymbol{p}}_i; \tilde{\boldsymbol{p}}_i; \bar{\boldsymbol{p}}_i - \tilde{\boldsymbol{p}}_i; \bar{\boldsymbol{p}}_i \odot \tilde{\boldsymbol{p}}_i] \tag{4.5}$$

$$\boldsymbol{t}_j = [\bar{\boldsymbol{h}}_j; \tilde{\boldsymbol{h}}_j; \bar{\boldsymbol{h}}_j - \tilde{\boldsymbol{h}}_j; \bar{\boldsymbol{h}}_j \odot \tilde{\boldsymbol{h}}_j] \tag{4.6}$$

where the symbols ; and $-$ respectively denote vector concatenation and elementwise subtraction, while $\odot$ denotes elementwise product. The new vector representations $\boldsymbol{s}_i$ and $\boldsymbol{t}_j$ are conveniently output as matrices $\boldsymbol{S} \in \mathbb{R}^{I \times k}$ and $\boldsymbol{T} \in \mathbb{R}^{J \times k}$ for all words in the premise and hypothesis respectively, where k is the dimensionality of each vector representation.

Similar to the *input encoding* stage, the *inference composition* accepts as input the output of the previous stage, $\boldsymbol{S}$ for the premise and $\boldsymbol{T}$ for the hypothesis, and then the input sequences are processed with BiLSTM units to yield new representations $\bar{\boldsymbol{s}}_i$ and $\bar{\boldsymbol{t}}_j$, which are output as matrices $\bar{\boldsymbol{S}} \in \mathbb{R}^{I \times 2k}$ and $\bar{\boldsymbol{T}} \in \mathbb{R}^{J \times 2k}$ for all words in the premise and hypothesis respectively:

$$\bar{\boldsymbol{S}}\ , \overrightarrow{\boldsymbol{c}}_s, \overleftarrow{\boldsymbol{c}}_s = BiLSTM(\boldsymbol{S}) \tag{4.7}$$

$$\bar{\boldsymbol{T}}\ , \overrightarrow{\boldsymbol{c}}_t, \overleftarrow{\boldsymbol{c}}_t = BiLSTM(\boldsymbol{T}) \tag{4.8}$$

where $\overrightarrow{\boldsymbol{c}}_s, \overleftarrow{\boldsymbol{c}}_s \in \mathbb{R}^k$ are respectively the last cell states in the forward and reverse directions of the BiLSTM that reads the premise. $\overrightarrow{\boldsymbol{c}}_t, \overleftarrow{\boldsymbol{c}}_t \in \mathbb{R}^k$ are similarly defined for the hypothesis.

The new representations are then max- and average-pooled elementwise to obtain a single vector representation:

$$\begin{aligned} \bar{\boldsymbol{s}}_{\text{ave}} &= \sum_{i=1}^{I} \frac{\bar{\boldsymbol{s}}_i}{I}, \quad \bar{\boldsymbol{s}}_{\max} = \max_{i=1}^{I} \bar{\boldsymbol{s}}_i \\ \bar{\boldsymbol{t}}_{\text{ave}} &= \sum_{j=1}^{J} \frac{\bar{\boldsymbol{t}}_j}{J}, \quad \bar{\boldsymbol{t}}_{\max} = \max_{j=1}^{J} \bar{\boldsymbol{t}}_j \\ \mathbf{x} &= \left[\bar{\boldsymbol{s}}_{\text{ave}}; \bar{\boldsymbol{s}}_{\max}; \bar{\boldsymbol{t}}_{\text{ave}}; \bar{\boldsymbol{t}}_{\max}\right] \end{aligned} \tag{4.9}$$

The vector $\mathbf{x}$ is then passed through a Multi-Layer Perceptron (MLP) classifier with a softmax output layer to make predictions:

$$\boldsymbol{y} = \text{softmax}\Big(\mathbf{W}_y\Big(\sigma_h\left(\mathbf{W}_h\mathbf{x} + \mathbf{b}_h\right)\Big) + \mathbf{b}_y\Big) \tag{4.10}$$

where $\mathbf{W}_y, \mathbf{W}_h$ are parameter matrices, $\mathbf{b}_y, \mathbf{b}_h$ are bias vectors, and σ_h is an activation function. Note that while the SNLI and MultiNLI datasets require three-way classification, our task requires two-way classification since we have only positive and negative classes.

ESIM with Conditional Encoding

We make the following adaptations to the model architecture of *ESIM*, which we found to be beneficial in the low-data regime when supervision is limited. When used for relation classification, *ESIM* encodes the sentence independently of the relation description. Given a new target relation's description, we want representations computed for the sentence to take into account the representations for the target relation description. In order to achieve this, we explore the use of conditional encoding, proposed for the task of textual entailment using LSTMs by Rocktäschel et al. (2016), and extended to BiLSTMs by Augenstein et al. (2016). The idea of conditional encoding for pairwise sequence classification tasks is to initialize the starting memory cell state when processing a sequence with the final memory cell state obtained from processing the other sequence. In our case we implement conditional encoding for BiLSTMs (cBiLSTM) by initializing the forward and reverse memory cell states of the BiLSTM that reads each sentence with the last forward and reverse memory cell states of the BiLSTM that reads the relation description. This was done for both the input encoding and inference composition stages of ESIM. Thus, Equations 4.1 and 4.7 can be expressed as Equations 4.11 and 4.12 respectively:

$$\bar{\boldsymbol{P}} = cBiLSTM(\boldsymbol{P}, \overrightarrow{\boldsymbol{c}}_h, \overleftarrow{\boldsymbol{c}}_h) \tag{4.11}$$

$$\bar{\boldsymbol{S}} = cBiLSTM(\boldsymbol{S}, \overrightarrow{\boldsymbol{c}}_t, \overleftarrow{\boldsymbol{c}}_t) \tag{4.12}$$

We refer to the adapted *ESIM* as the Conditioned Inference Model (*CIM*) in subsequent sections.

4.2.2 Dataset

We evaluate our approach using datasets from Adel et al. (2016) and Levy et al. (2017). The dataset of Adel et al. (2016) (*LMU-RC*) is split into training, development and evaluation sets. While the training split was generated by distant supervision, the development and test data were obtained from manually annotated TAC-KBP system outputs. We obtained the descriptions for the relations from the TAC-KBP relation ontology guidelines.[2] This turns each instance in the dataset into a tuple consisting of a relation, its subject and object entities, a sentence containing both entities and a relation description.

We applied a similar process to the dataset released by Levy et al. (2017) (*UW-RE*), which was derived from the WikiReading dataset (Hewlett et al.,

[2] https://tac.nist.gov/2015/KBP/ColdStart/guidelines/TAC_KBP_2015_Slot_Descriptions_V1.0.pdf

Dataset	Model	F1 (%)
LMU-RC	ESIM	20.16
	CIM	**22.80**
UW-RE	ESIM	61.32
	CIM	**64.78**

Table 4.2 Zero-shot relation learning results for *ESIM* and *CIM* using Distant Supervision (DS) data.

Dataset	Supervision	F1 (%)
LMU-RC	TE	25.54
	TE+DS	**26.28**
UW-RE	TE	44.38
	TE+DS	**62.33**

Table 4.3 Zero-shot relation learning results for model CIM pre-trained on two sources of data: Textual Entailment (TE), or both Distant Supervision and Textual Entailment (TE+DS).

2016). It consists of 120 KB relations and a set of question templates for each relation, containing both positive and negative relation instances, with each instance consisting of a subject entity, a knowledge base relation, a question template for the KB relation, and a sentence retrieved from the subject entity's Wikipedia page. We wrote descriptions for each of the 120 relations in the dataset, with each relation's question templates serving as a guide. As an example, given the KB relation *director(X, Y)*, its associated question template *Who is the director of X?* is converted to a relation description of the form *The director of X is Y*. Thus all instances in the dataset now include the corresponding relation description, making them suitable for the task of zero-shot relation classification using our approach. [3]

In addition to the two datasets, we also utilize the *MultiNLI* natural language inference corpus (Williams et al., 2017) in our experiments as a source of supervision. We map its *entailment* and *contradiction* class instances to positive and negative relation instances respectively.

4.2.3 Experiments and Results

We conduct two series of experiments. The first set of experiments tests the performance of our approach in the zero-shot setting, where no supervision instances are available for new relations. The second set of experiments

[3] This conversion was performed by the author.

measure the performance of our approach in the limited supervision regime, where varying levels of supervision is available.

Implementation and Hyperparameters

Our approach is implemented with Tensorflow (Abadi et al., 2016). We initialize the word embedding layer with 300D Glove (Pennington et al., 2014) vectors and apply Dropout with a keep probability of 0.9 (this value is used for all experiments and not tuned for any particular individual experiment) to all layers. The result reported for each experiment is the average taken over five runs with different random seeds. In order to prevent overfitting to specific entities, we mask out the subject and object entities with the tokens *SUBJECT_ENTITY* and *OBJECT_ENTITY* respectively.

Zero-shot Relation Learning

For this experiment we created ten folds of each dataset, with each fold partitioned into train/dev/test splits along relations. In each fold, a relation belongs exclusively to either the train, dev or test splits.

Table 4.2 shows averaged F1 across the folds for the models on the *LMU-RC* and *UW-RE* datasets using their Distant Supervision (DS)-generated training data. We observe that even without training data for the test relations, the models were still able to make predictions for them, though at different performance levels. *CIM* obtained better performance compared to *ESIM*, as a result of its use of conditional encoding.

Table 4.3 shows F1 scores of model *CIM* pre-trained on only MultiNLI (TE) or a combination of MultiNLI and distant supervision (TE+DS) data in the zero-shot setting. We find that *CIM* pre-trained on only textual entailment data is already able to make predictions for unseen relations, while using a combination of distant supervision and textual entailment data achieved improved F1 scores across both datasets, demonstrating the validity of our approach in this setting. We also observe that using TE+DS data performs worse than using DS data alone in the case of the *UW-RE* dataset, unlike in the case of *LMU-RC*. This is possibly because DS data performs much better for the former.

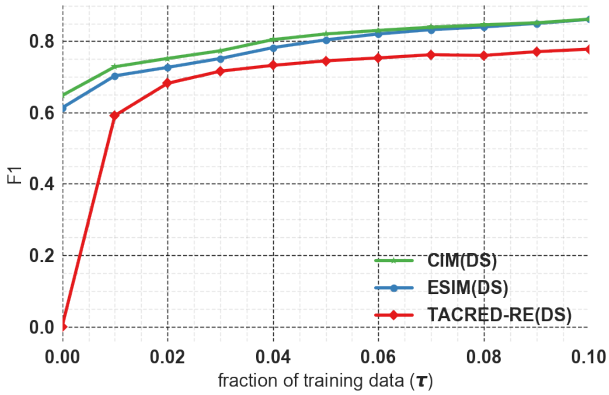


Figure 4.1 Limited supervision results: F1 scores on *UW-RE* as fraction of training data (τ) is varied. When τ=0, we get the zero-shot results in Table 4.2

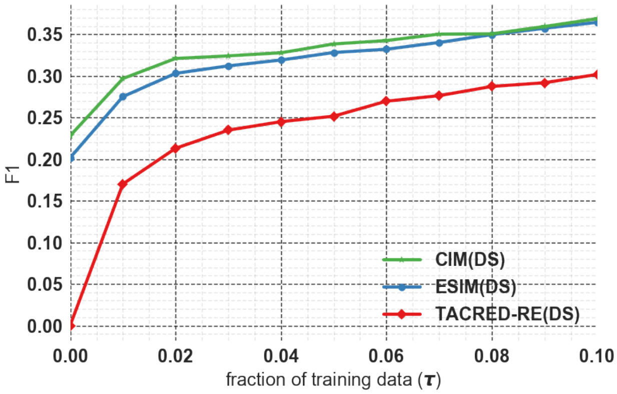


Figure 4.2 F1 scores on *LMU-RC* as fraction of training data (τ) is varied.

Limited Supervision Relation Learning

For the experiments in this limited-supervision setting, we randomly partition the dataset along relations into a train/dev/test split. Similar to the zero-shot setting, a relation belongs to each split exclusively. Then for each experiment, we make available to each model a fraction τ of example instances of unobserved relations as supervision. Note that the particular example instances used are a disjoint set of instances which are not present in the development and evaluation sets.

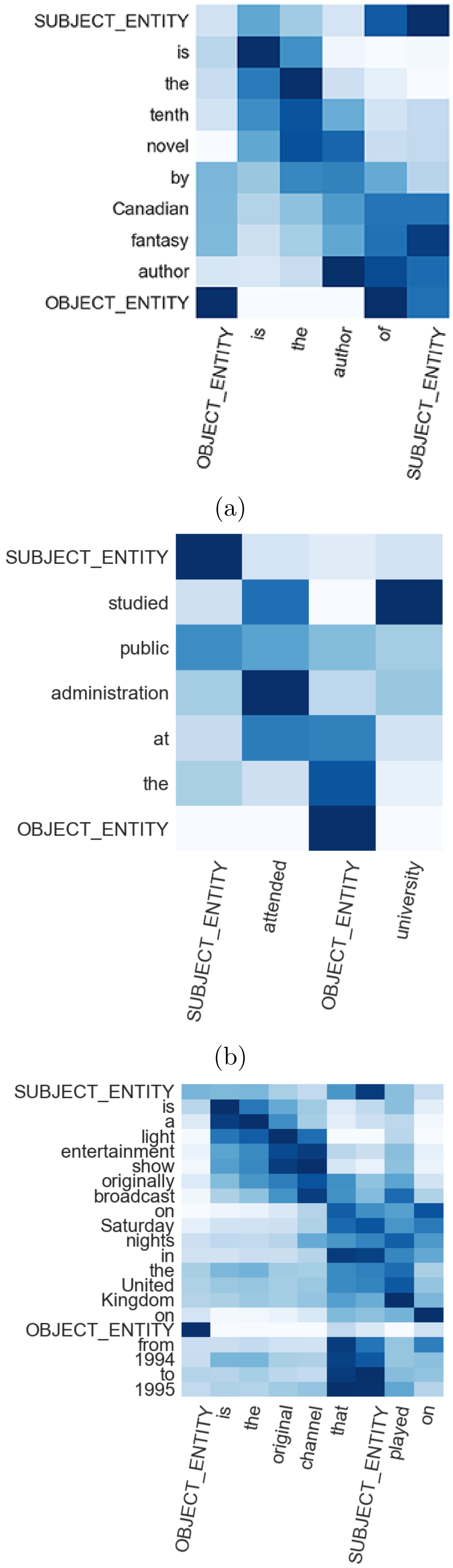
In addition to ESIM and the proposed CIM, we also report results for the TACRED Relation Extractor (*TACRED-RE*), the position-aware RNN model that was found to achieve state-of-the-art results on the TACRED (Zhang et al., 2017) dataset. *TACRED-RE* is a supervised model that expects labelled data for all relations during training, and thus not applicable in the zero-shot setup.

Results for this set of experiments are shown in Figure 4.1 for the *UW-RE* dataset. We find that only about 5% of the training data is required for both *ESIM* and *CIM* to reach around 80% in F1 performance, with *CIM* outperforming *ESIM* in the 0-6% interval. However, beyond this interval, we do not observe any major difference in performance between ESIM and *CIM*, demonstrating that CIM performs well in both the zero-shot and limited supervision settings. For context, when given full supervision on the *UW-RE* dataset, *CIM* and *TACRED-RE* obtain F1 scores of 94.82% and 87.73% respectively. A similar trend is observed for the *LMU-RC* dataset, whose plot can be found in Figure 4.2.

In general, all models obtain better results on *UW-RE* than on *LMU-RC*. We hypothesize that the performance difference is due to *UW-RE* being derived from Wikipedia documents (which typically have well-written text), while LMU-RC was obtained from different genres and sources (such as discussion forum posts and web documents), which tend to be noisier.

Qualitative Results

Figure 4.3 depicts a visualization of the normalized attention weights assigned by our model on randomly drawn instances from the development set. We observe that it is able to attend to words that are semantically coherent with the premise (“novel” and “author”, Figure 4.3a), (“studied” and “university”, Figure 4.3b), (“show” and “channel”, Figure 4.3c).



(a)

(b)

(c)

Figure 4.3 Attention visualization

4.3 Related Work

Recent work, including Adel et al. (2016) and Zhang et al. (2017), proposed models that assume the availability of supervised data for the task of relation classification. Adel et al. (2016) conducted a study that compared the effectiveness of Convolutional Neural Networks (CNNs) to other models for relation classification. For their study, they made use of a dataset derived from the TAC-KBP tasks. They report that the different models performed well on a different subsets of the relation slots, and that each relation has specific properties that make particular models suitable for extracting them. Zhang et al. (2017) proposed a model and dataset for relation classification. Their model makes use of an attention mechanism that takes into consideration the position of the tokens in the sentence relative to the position of the subject and object entities. They also derived a relation classification dataset from the TAC-KBP tasks. However, a major shortcoming common to the proposed approaches by Adel et al. (2016) and Zhang et al. (2017) is that they identify only relations observed at training time, and are unable to generalize to new (unobserved) relations at test time. Our work proposes a way to address this weakness by using the natural language description of relations paired with a textual entailment model.

Levy et al. (2017) showed that the task of slot filling in relation extraction can be reduced to a question answering problem. The task we address in this work is that of zero-shot *relation classification*, which determines if a given relation exists between two given entities in text. As a result the output of our approach is a binary classification decision indicating whether a given relation exists between two given entities in text, while Levy et al. (2017) returns the span corresponding to the relation slot ("answers") from the text.

Our approach is also inspired by recent methods for leveraging knowledge from a set of source tasks to target tasks, such as recent transfer learning methods in natural language processing (Peters et al., 2018; McCann et al., 2017). Peters et al. (2018) and McCann et al. (2017) show that representations learned from language modelling and machine translation respectively can enhance performance when transferred to a number of other natural language tasks. Both approaches did not consider the task of relation classification, and moreover, our work utilizes relation descriptions within the framework of textual entailment to enable zero-shot relation classification.

4.4 Summary

In this chapter we showed how relation classification can be formulated as a textual entailment task, and that this leads to several advantages. In contrast to previous relation classification models, we were able to perform zero-shot classification of relations through the use of relation descriptions and use existing textual entailment models and datasets to perform relation classification. We performed experiments on two datasets, and demonstrated the effectiveness of our approach in both quantitative and qualitative results.

The approaches we have proposed in this chapter relies on the use of a textual entailment model. However, a number of models have been proposed in the literature specifically for relation classification. In the next chapter, we propose a way to improve the performance of these relation classification models when trained with limited supervision.

Chapter 5

Model-Agnostic Meta-Learning for Relation Classification

The previous chapters proposed approaches to relation classification with improved predictive performance, both in settings with limited supervision instances and in settings with zero supervision instances for new relations. However these previous approaches assume the use of specific types of model architectures. The implication of this is that the previously proposed approaches may not be directly applicable when a different model architecture is used.

However, there is a wide range of other model architectures that have been proposed for relation classification, especially supervised models based on neural networks, for instance recursive neural networks (Socher et al., 2012; Hashimoto et al., 2013), convolutional neural networks (Zeng et al., 2014; Nguyen and Grishman, 2015), recurrent neural networks (Zhang and Wang, 2015; Xu et al., 2015; Zhang et al., 2017) or some combination of recurrent and convolutional neural networks (Vu et al., 2016). These models are typically highly data inefficient, requiring significant quantities of supervision data to generalize well. A relation classification approach that reduces the amount of supervision required by these models, and that does not adversely affect their performance in the process, is therefore highly desirable.

In this chapter we aim to address this shortcoming, in order to move beyond our previously proposed approaches for relation classification with limited supervision, by proposing a model-agnostic approach to enhance the predictive performance of a wide range of other relation classification models. We explore a more general approach for training relation classification models in settings with limited supervision, which is applicable to any existing gradient-optimized relation classification model.

Motivated by the observation that meta-learning leads to learning a better parameter initialization for new tasks than ad hoc multi-task learning across all tasks (Finn et al., 2017), we frame the task of supervised relation classification as an instance of meta-learning. By leveraging gradient-based meta-learning, we propose a model-agnostic meta-learning protocol for training relation classifiers to achieve enhanced predictive performance in limited supervision settings. During training, our algorithm aims to not only learn good parameters for classifying relations with sufficient supervision, but also learn model parameters that can be fine-tuned to enhance predictive performance for relations with limited supervision. We conduct experiments on two relation classification datasets, and demonstrate that the proposed meta-learning approach improves the predictive performance of two state-of-the-art supervised relation classification models, the position-aware relation classification model proposed in Zhang et al. (2017) (*TACRED-PA*) and the contextual graph convolution networks proposed in Zhang et al. (2018) (*C-GCN*), with varying amounts of supervision available at training time.

The rest of this chapter is structured as follows. We begin by providing background on meta-learning in Section 5.1. We define meta-learning and discuss the various approaches that have been proposed for it in the literature. Next, we describe how we apply meta-learning for relation classification, and provide a model-agnostic metal-learning procedure for training relation classification models to enhance their predictive performance in limited supervision settings in Section 5.2. In Section 5.3 we report and discuss the results of experiments conducted using two state-of-the-art relation classification models on two datasets, and Section 5.5 concludes with a summary.

5.1 Background

Meta-learning, also known as *learning to learn* (Thrun and Pratt, 1998), aims to develop models and algorithms which are able to exploit background knowledge to adaptively improve their learning process with experience. A number of meta-learning approaches have been proposed, and broadly fall into the three following lines of work: learning how to update model parameters from background knowledge (for instance, Andrychowicz et al. 2016; Ravi and Larochelle 2017), specific model architectures for learning with limited supervision (for instance, Vinyals et al. 2016; Snell et al. 2017), and model-agnostic methods for learning a good parameter initialization for learning with limited supervision (for instance, Finn et al. 2017; Nichol et al. 2018).

We next give a brief overview of the model-agnostic methods for meta-learning, which learn a good parameter initialization for target tasks from a set of source tasks, as proposed in Finn et al. (2017) and Nichol et al. (2018). These algorithms work by training a meta-model on the set of source tasks, such that the meta-model provides a good parameter initialization for target tasks which are taken from the same distribution as the source tasks. At test time, such an initialization can be fine-tuned with a limited number of gradient steps using a limited amount of training examples from the target tasks, in order to achieve good performance on the target tasks.

In formal terms, let $p(\mathcal{T})$ be the distribution over tasks and f_θ be the function learned by a neural model parametrized by θ. During adaptation to each task $\mathcal{T}_i$ sampled from $p(\mathcal{T})$, the model parameters θ are updated to task-specific parameters θ_i'. For a single gradient step, for instance, this update can be carried out as:

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta) \tag{5.1}$$

where $\mathcal{L}_{\mathcal{T}_i}$ is the loss on task $\mathcal{T}_i$ and α is the step size hyperparameter.

The model parameters θ are trained to optimize the performance of $f_{\theta_i'}$, after taking a number of gradient steps with limited example instances from tasks sampled from $p(\mathcal{T})$. This can be achieved by utilizing the meta-objective:

$$\min_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}) = \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)}) \tag{5.2}$$

The optimization of the meta-objective is performed across tasks using *SGD*, by making updates to θ:

$$\theta \leftarrow \theta - \epsilon \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}) \tag{5.3}$$

where ϵ is the meta step size parameter.

Intuitively, the meta-objective explicitly encourages the model to learn model parameters that can be quickly adapted to achieve optimum predictive performance across all tasks using a few gradient descent steps. In the next section, we describe how we apply this framework to improve performance of relation classification models with limited supervision.

5.2 Model-Agnostic Meta-Learning for Relation Classification

If we consider each relation $\mathcal{R}_i$ as a task, then one approach to supervised relation classification with limited supervision is to directly train a multi-class classifier for all relations jointly. For all relations $\mathcal{R}_i$ from a distribution $p(\mathcal{R})$, this approach directly optimizes for the following objective:

$$\theta^* = \min_{\theta} \sum_{\mathcal{R}_i \sim p(\mathcal{R})} \mathcal{L}_{\mathcal{R}_i}(f_\theta) \tag{5.4}$$

where $\mathcal{L}_{\mathcal{R}_i}$ is the loss on relation $\mathcal{R}_i$. This assumes that joint training on all relations would naturally result in the optimal model parameters θ^* with good predictive performance for all relations. This is however not necessarily the case, especially for relations with limited training instances from which the model can learn to generalize.

We propose to instead utilize meta-learning to explicitly encourage the model to learn a good joint parameter initialization for all relations, which can then be fine-tuned with limited supervision from each relation's training instances to achieve good performance on its test set. Such parameter initializations would be especially beneficial for enhancing performance on relations with limited training instances.

Observe though that directly optimizing Equation 5.2 requires computing second order derivatives over the parameters, which can be computationally expensive. Thus, we follow Nichol et al. (2018) by approximating the meta-objective in Equation 5.2 with the training Algorithm in 2. Though the Model-Agnostic Meta-Learning (MAML) algorithm of Finn et al. (2017) also has a first-order variant known as First-Order MAML (FOMAML), here we use the algorithm of Nichol et al. (2018) (REPTILE) for simplicity.

Subsequently we refer to our overall training procedure as summarized in Algorithm 2 as Meta-learning Relation Classification (*MLRC*). We assume access to f_θ (learner model), which is a relation classification model parameterized by θ and a distribution over relations $p(R)$. The algorithm consists of the meta-learning phase (lines 1-10), followed by the supervised learning phase (line 11) which fine-tunes the meta-learned parameters, both carried out on a relation classification model using the same data for both stages.

In the first phase of learning, each iteration in our approach starts by sampling a batch of relations from $p(R)$ (line 3). Then for each relation

Algorithm 2 Meta-Learning Relation Classification (*MLRC*)

```
Require: distribution over relations $p(\mathcal{R})$
Require: relation classification function $f_\theta$
Require: gradient-based optimization algorithm (e.g. $SGD$)
Require: step size $\epsilon$, learning rate $\alpha$
 1: randomly initialize $\theta$
 2: while not done do
 3:     Sample batch of $\mathcal{B}$ relations $\mathcal{R}_i \sim p(\mathcal{R})$
 4:     for all $\mathcal{R}_i$ do
 5:         Sample train instances $\mathcal{D} = \{x^{(j)}, y^{(j)}\}$ from $\mathcal{R}_i$
 6:         Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{R}_i}(f_\theta)$ using $\mathcal{D}$
 7:         Compute adapted parameters:
            $\theta'_i = SGD(\theta_i, \nabla_\theta \mathcal{L}_{\mathcal{R}_i}(f_\theta), \alpha)$
 8:     end for
 9:     Compute update of meta-parameters:
        $\theta = \theta - \epsilon \frac{1}{\mathcal{B}} \sum_{i=1}^{i=\mathcal{B}} (\theta'_i - \theta)$
10: end while
11: Fine-tune $f_\theta$ with standard supervised learning.
```

we sample a batch of supervision instances $\mathcal{D}$ from its training set (line 5). We then obtain the adapted model parameters θ'_i on this relation by first computing the gradient of the training loss using the sampled relation instances (line 6) an updating the parameters with a gradient-based optimization algorithm such as *SGD* or *Adagrad* (Duchi et al., 2011) (line 7). At the end of the learning iteration, the adapted parameters on each sampled relation in the batch are averaged, and an update is made on the model parameters θ (line 9).

In the second phase of learning, we first initialize the model parameters with that learned during meta-training. We then proceed to fine-tune the model parameters with standard supervised learning by taking a number of gradient descent steps using the same randomly sampled batches of supervision instances from the relations' training set as was used during meta-learning (line 11).

5.3 Experiments

5.3.1 Relation Classification Models

We adopt as the learner model (f_θ) two recent supervised relation classification models, the position-aware model of Zhang et al. (2017) (*TACRED-PA*) and the contextual graph convolution networks proposed in Zhang et al. (2018)

(*C-GCN*), both of which are multi-class models with parameters optimized via stochastic gradient descent.

5.3.2 Setup

We conduct experiments in a limited supervision setting, where we provide all models with the same fraction of randomly sampled supervision instances during training. Further, for each experiment the supervision instances within each fraction is exactly the same across all models. We report results for each experiment by taking the average over ten (10) different runs.

5.3.3 Datasets

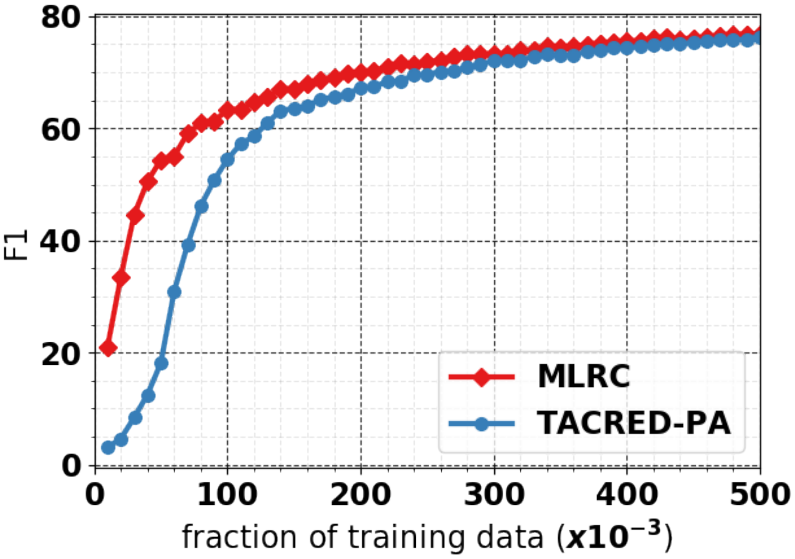
We evaluate our approach on the SemEval-2010 Task 8 relation classification dataset (Hendrickx et al., 2009) (*SemEval*), and on the TACRED dataset (Zhang et al., 2017) (*TACRED*). The *SemEval* dataset has a total of 8000 training and 2717 testing instances respectively. For experiments the training set is split into two, and we use 7500 instances for training and 500 instances for development. For *TACRED*, we use the standard training, development and testing splits as provided by Zhang et al. (2017).

5.3.4 Experimental Details and Hyperparameters

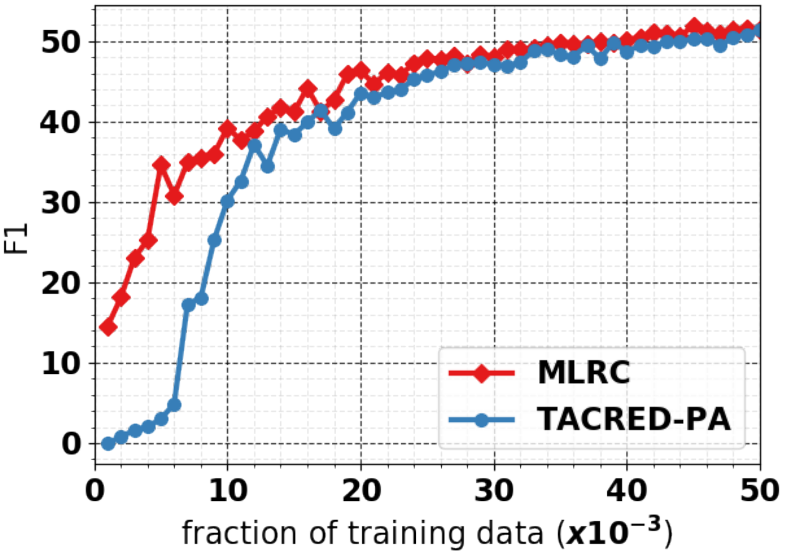
We initialize word embeddings with Glove vectors (Pennington et al., 2014) and did not fine-tune them during training. Model training and parameter tuning are carried out on the training and development splits of each dataset, and final results reported on the test set.

We ensure all models have access to the same data. For model *MLRC*, for each fraction, we train for 150 meta-learning iterations on *TACRED* dataset and 1000 meta-iterations on the *SemEval* dataset using that fraction of data. We then fine-tune with standard supervised learning using exactly the same data as was used during meta-learning.

For both relation classification models, that is *TACRED-PA* and *C-GCN*, we use the same hyperparameters as in Zhang et al. (2017) and Zhang et al. (2018) respectively.

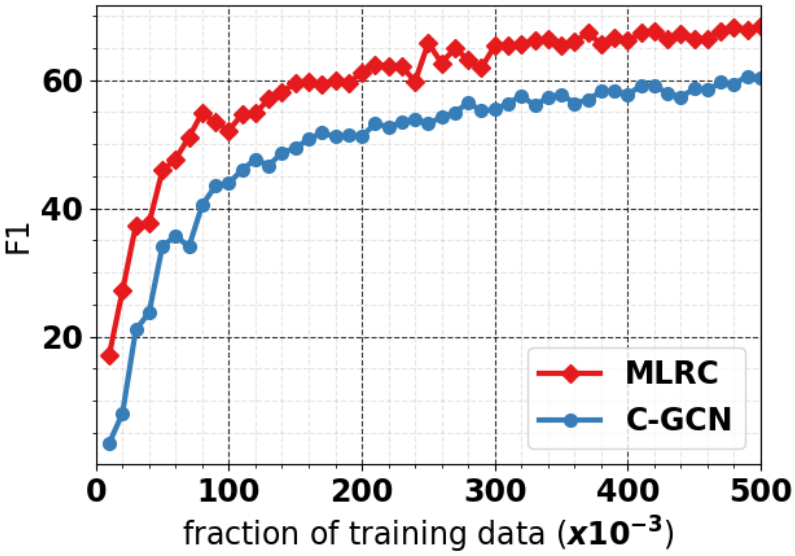


(a)

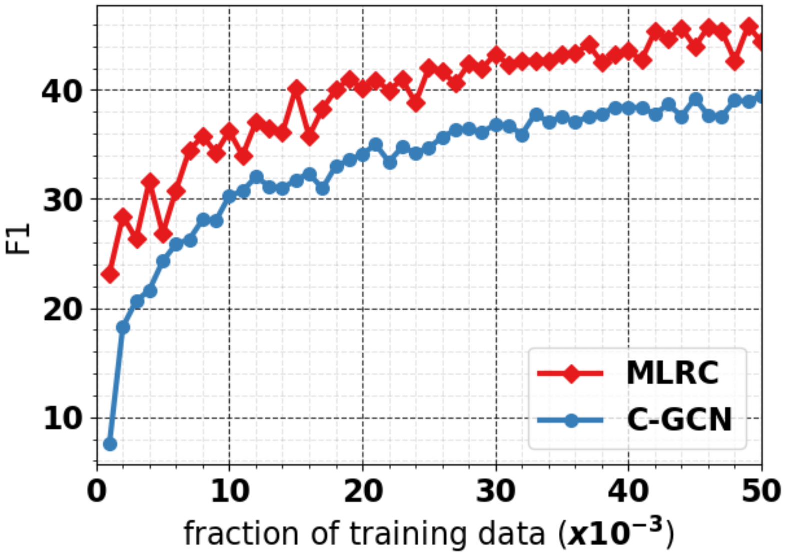


(b)

Figure 5.1 Results obtained using *TACRED-PA* as the learner model on (a) *SemEval*, and (b) *TACRED* datasets



(a)



(b)

Figure 5.2 Results obtained using *C-GCN* as the learner model on (a) *SemEval*, and (b) *TACRED* datasets

Relation	#	F1(%)	
		TC-PA	MLRC
Instrument-Agency	3	0	8.44
Content-Container	4	0.93	30.9
Member-Collection	5	3.04	24.19
Entity-Destination	7	14.33	35.36
Entity-Origin	7	2.85	24.62
Message-Topic	7	0.8	12.32
Component-Whole	8	2.68	14.87
Product-Producer	9	0.68	10.29
Cause-Effect	11	2.93	28.52
Average		3.13	21.05

Table 5.1 Results with 1% training data on *SemEval* using *TACRED-PA* as the learner model. The # column is the number of instances of each relation during training, and TC-PA denotes the *TACRED-PA* model (trained without meta-learning), while *MLRC* denotes the same model trained with our approach.

Relation	#	F1(%)	
		C-GCN	MLRC
Instrument-Agency	4	0	21.83
Member-Collection	4	4.53	23.81
Cause-Effect	5	4.52	12.84
Component-Whole	5	3.72	8.91
Message-Topic	5	0.32	3.23
Content-Container	6	2.35	31.98
Entity-Origin	6	2.42	5.74
Entity-Destination	9	10.38	36.05
Product-Producer	12	2.21	9.83
Average		3.38	17.14

Table 5.2 Results with 1% training data on *SemEval* using *C-GCN* as the learner model. The # column is the number of instances of each relation during training, and *C-GCN* is the *C-GCN* model trained without meta-learning, while *MLRC* denotes the same model trained with our approach.

5.3.5 Evaluation Metrics

For the *TACRED* dataset, we follow Zhang et al. (2017) and report micro-averaged F1 scores. We use the same evaluation script as Zhang et al. (2017) for computing this metric. For the *SemEval* dataset, we report the official measure, which is the F1 score macro-averaged across relations, which we compute using the official evaluation script that comes with the dataset.

5.3.6 Results and Discussion

The results obtained on the *SemEval* and *TACRED* datasets using *TACRED-PA* as the learner model (f_θ) are shown in Figures 5.1a and 5.1b respectively. We find that on both datasets, our approach improves performance as more supervision becomes available, with the largest gains obtained at the early stage when very limited supervision is available. For instance on *SemEval*, given just 1% of the training set (first datapoint in Figure 5.1a), our approach improves the F1 performance of *TACRED-PA* from 3.13% to 21.05%, representing an absolute increase of 17.92%. Table 5.1 gives a further breakdown of the F1 scores of individual relations when both approaches are given access to 1% of the training set. We observe that *MLRC* considerably improves the performance of *TACRED-PA* on relations with the least number of training instances, likely by leveraging background knowledge from relations with more training instances. On the *TACRED* dataset, *MLRC* improves the performance of *TACRED-PA* from 2.98% to 34.59% with just 0.5% of the training data (fifth datapoint in Figure 5.1b), which is an absolute increase of 31.61%.

A similar trend is observed using *C-GCN* as the learner model on both datasets, as presented in Figures 5.2a and 5.2b. For instance on *SemEval*, we improve the F1 performance of *C-GCN* from 3.38% to 17.14% using just 1% of the training data (first datapoint in Figure 5.2a). Table 5.2 gives a further breakdown of the F1 scores of individual relations when both approaches are given access to 1% of the training set. Similarly on *TACRED*, the performance of *C-GCN* is improved from 7.59% to 23.18% (first datapoint in Figure 5.2b) by using 0.1% of its training set.

Further, we find that the proposed approach does not adversely affect performance when full supervision is available during training. For instance, when given full supervision on the *TACRED* dataset, while *TACRED-PA* obtains an F1 score of 65.1%, its performance is improved to 65.2% by using our approach, demonstrating that the proposed approach does not adversely affect performance when provided full supervision during training.

5.4 Related Work

In recent work, Han et al. (2018) proposed a dataset and evaluation setup for few-shot relation classification which assumes access to full supervision for training relations (specifically 700 instances per relation). In contrast, we address a different setting in which only limited supervision is available for all relations. In addition, the setup in Han et al. (2018) requires a model architecture *specific* to few-shot learning based on distance metric learning. On the other hand, the approach proposed in this chapter has the advantage that it applies to any gradient-optimized relation classification model.

5.5 Summary

In this chapter, we show that the performance of supervised relation classification models can be improved, even with limited supervision at training time, by framing relation classification as an instance of meta-learning, and proposed a model-agnostic learning protocol for training relation classifiers with enhanced predictive performance in limited supervision settings. We demonstrate the effectiveness of this approach using two state-of-the-art neural relation classification models on two relation classification datasets, in all cases improving performance when limited supervision instances is available for training relation classifiers.

The methods we have proposed up to this chapter assume that all supervision data is available at the start of training, and to incorporate new supervision data after training will require substantial retraining. This can be computationally expensive. For instance, the approach presented in this chapter requires more total training time compared to only performing supervised training, since it involves both a meta-training and supervised training phase. In the next chapter we mitigate this problem, and present an algorithm that is able to continually incorporate new supervision data as it becomes available, without the need for substantial retraining.

Chapter 6

Lifelong Relation Classification with Meta-Learning

The model-agnostic approach proposed in the last chapter for relation classification is general in the sense that it applies to any gradient-optimized relation classification model. One important assumption of the approach is that supervision for all relations is available before training commences. This implies that the approach is unable to detect an evolving set of novel relations observed after training without substantial retraining the model, which can be computationally expensive and may lead to catastrophic forgetting of previously learned relations.

This problem can be partially addressed by zero-shot relation classification approaches, such as the approach proposed in Chapter 4, which are able to classify at test time relations that are not seen at training time. However, though the zero-shot relation classification approaches can classify unseen relations, their zero-shot performance is lower when compared to their performance for seen relations, and are unable to continually exploit any newly available supervision to improve performance without considerable retraining. It is thus desirable for relation classification models to have the ability to continually incorporate newly available supervision, in order to enable them utilize new supervision as it becomes available to both improve their performance on previously known relations and to enable them classify new relations.

One approach to tackle this problem was proposed by Wang et al. (2019), who introduced an embedding alignment approach to enable continual learning for relation classification models. They consider a setting with streaming tasks, where each task consists of a number of distinct relations, and proposed to align the representation of relation instances in the embedding space to enable continual learning of new relations without forgetting knowledge from past relations. While they obtained promising results, a key weakness of their approach is that the use of an alignment model introduces additional parameters to already over-parameterized relation classification models, which may in turn lead to an increase in the quantity of supervision required for training. In addition, their approach can only align embeddings between observed relations, and does not have any explicit objective that encourages the model to transfer and exploit knowledge gathered from previously observed relations to facilitate the efficient learning of yet to be observed relations.

This chapter investigates and present results for an approach that gives relation classification models the ability to continually learn without forgetting from new supervision data, based on a combination of ideas from lifelong learning and optimization-based meta-learning. We propose to consider lifelong relation classification as a meta-learning challenge, to which the machinery of current optimization-based meta-learning algorithms can be applied. Unlike the use of a separate alignment model as proposed in Wang et al. (2019), our approach does not introduce additional parameters. In addition, our proposed approach is more data efficient since it explicitly optimizes for the transfer of knowledge from past relations, while avoiding the catastrophic forgetting of previously learned relations. Empirically, we evaluate on lifelong versions of the datasets by Bordes et al. (2015) and Han et al. (2018) and demonstrate considerable performance improvements over prior state-of-the-art approaches.

The remainder of this chapter is structured as follows. Section 6.1 starts by defining and contrasting lifelong learning and meta-learning, and gives an outline of recent research directions in both learning setups in order to provide motivation for our proposed approach. Thereafter, Section 6.2 describes our proposed algorithm for lifelong relation classification with meta-learning which gives relation classification models the ability to utilize new supervision as it becomes available, without the need for substantial retaining and without forgetting knowledge of how to detect previous relations. In Section 6.4 we discuss results obtained on experiments conducted on two lifelong relation classification benchmarks. Finally, Section 6.5 concludes with a summary.

6.1 Background

In this section we provide relevant background work on lifelong learning, and then contrast it with recent works in gradient-based meta-learning. This serves to provide motivation for our main hypothesis in this chapter, which is that complementary gains can be obtained from the synthesis of ideas from the two learning settings for building relation classification models that are able to learn continually with limited supervision, without forgetting knowledge of past relations.

In the lifelong learning setting, also referred to as continual learning (Ring, 1994; Thrun, 1996; Zhao and Schmidhuber, 1996), a model f_θ is presented with a sequence of tasks $\{\mathcal{T}_t\}_{t=1,2,3..,T}$, one task per round, and the goal is to learn model parameters $\{\theta_t\}_{t=1,2,3,..,T}$ with the best performance on the observed tasks. Each task $\mathcal{T}$ can be a conventional supervised task with its own distinct train ($\mathcal{T}^{train}$), development ($\mathcal{T}^{dev}$) and test ($\mathcal{T}^{test}$) splits. At each round t, the model is allowed to exploit knowledge gained from the previous $t-1$ tasks to enhance performance on the current task. In addition, the model is also allowed to have a small-sized buffer memory B, which can be used to store a limited amount of data from previously observed tasks.

A prominent line of work in lifelong learning research is developing approaches that enable models learn new tasks without forgetting knowledge from previous tasks, i.e. avoiding catastrophic forgetting of old tasks (McCloskey and Cohen, 1989; Ratcliff, 1990; McClelland et al., 1995; French, 1999). Approaches proposed to address this problem include memory-based approaches (Lopez-Paz, David and Ranzato, 2017; Rebuffi et al., 2017; Chaudhry et al., 2018); parameter consolidation approaches (Kirkpatrick et al., 2017; Zenke et al., 2017); and dynamic model architecture approaches (Xiao et al., 2014; Rusu et al., 2016; Fernando et al., 2017).

In contrast to lifelong learning, meta-learning, or learning to learn (Schmidhuber, 1987; Naik and Mammone, 1992; Thrun and Pratt, 1998), aims to develop algorithms that learn a generic knowledge of how to solve tasks from a given distribution of tasks, by generalizing from solving related tasks from that distribution. While recent gradient-based meta-learning algorithms were proposed and evaluated in the context of few-shot learning, in this chapter we demonstrate their effectiveness when utilized in the lifelong learning setting for relation extraction. This follows similar intuition as recent work by Finn et al. (2019), who explored the usefulness of meta-learning for online (regret-based) learning.

6.2 Meta-Learning for Lifelong Relation Classification

It can be inferred from the previous section that a lot of lifelong learning research has focused on approaches to avoid catastrophic forgetting (i.e. negative backward transfer of knowledge) while recent meta-learning studies have focused on effective approaches for positive forward transfer of knowledge (for few-shot tasks). Given the complementary strengths of the approaches from the two learning settings, we propose to embed meta-learning into the lifelong learning process for relation classification.

As stated previously, given tasks $\mathcal{T}$ sampled from a distribution of tasks $p(\mathcal{T})$, and a learner model f_θ, gradient-based meta-learning methods learn a prior initialization of the parameters of the model which, at meta-test time, can be quickly adapted to achieve good performance on a new task using a few steps of gradient descent. During adaptation to the new task, the model parameters θ are updated to task-specific parameters θ' with good performance on the task. In formal terms, one view of gradient-based meta-learning algorithms is that they are optimizing the meta-objective:

$$\begin{aligned} &\min_\theta \mathbb{E}_{\mathcal{T}\sim p(\mathcal{T})}\left[\mathcal{L}_\mathcal{T}\left(\theta'\right)\right] = \\ &\min_\theta \mathbb{E}_{\mathcal{T}\sim p(\mathcal{T})}\left[\mathcal{L}_\mathcal{T}\left(\mathcal{U}\left(\mathcal{D}_\mathcal{T};\theta\right)\right)\right] \end{aligned} \tag{6.1}$$

where $\mathcal{L}_\mathcal{T}$ is the loss and $\mathcal{D}_\mathcal{T}$ is training data from task $\mathcal{T}$, and $\mathcal{U}$ is a gradient descent learning rule such as SGD. Note that $\mathcal{U}$ is not restricted to only vanilla SGD, and it can be any gradient descent learning rule. Thus, we can embed the meta-learning objective directly within lifelong learning for relation extraction.

Our algorithm for lifelong relation extraction is illustrated in Algorithm 3. We start by randomly initializing the parameters of the relation extraction model (the *learner*) (line 1). Then, as new tasks arrive, we augment their training set with randomly sampled task exemplars from the buffer memory B (lines 2-9). We then sample a batch of relations from the augmented training set (line 10). Then for each sampled relation $\mathcal{R}_i$, we sample a batch of supervision instances $\mathcal{D}^{train}_{\mathcal{R}_i}$ from its training set (line 11-12). We then obtain the adapted model parameters θ^i_t by first computing the gradient of the training loss using the sampled relation instances (line 13) and updating the parameters with a gradient-based optimization algorithm (such as *SGD* or *Adagrad* (Duchi et al., 2011)) (line 14). At the end of the learning iteration, the adapted parameters on all sampled relations in the batch are averaged, and an update is made on the task parameters θ_t (line 16). This is done until convergence on the current task, after which exemplars of the current task are added to the buffer memory (line 18). Task exemplars are obtained by first clustering all training instances of the current task into 50 clusters using K-Means, then selecting an instance from each cluster with a representation closest to the cluster prototype. Finally, the model parameters are updated to the current task's adapted parameters (line 19).

Algorithm 3 Meta-Learning for Lifelong Relation Extraction (*MLLRE*)

```
Require: Stream of incoming tasks $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, ...$
Require: Relation extraction function $f_\theta$
Require: Optimization algorithm (e.g. SGD)
Require: Step size $\epsilon$, learning rate $\alpha$
Require: Buffer memory $B$
 1: Randomly initialize $\theta$
 2: while there are still tasks do
 3:     Retrieve next task $\mathcal{T}_t$ from stream
 4:     Initialize $\theta_t \leftarrow \theta$
 5:     repeat
 6:         if $B$ is not empty then
 7:             Retrieve exemplars $\mathcal{E}$ of random task from $B$
 8:             Update task training set $\mathcal{D}_t^{train} = \mathcal{D}_t^{train} \cup \mathcal{E}$
 9:         end if
10:         Sample random relations $\{\mathcal{R}_i\}_{i=1}^{N}$ from $\mathcal{D}_t^{train}$
11:         for each $\mathcal{R}_i$ do
12:             Sample train instances $\mathcal{D}_{\mathcal{R}_i}^{train}$ of $\mathcal{R}_i$
13:             Evaluate $\nabla_{\theta_t} \mathcal{L}_{\mathcal{R}_i}(f_{\theta_t})$ using $\mathcal{D}_{\mathcal{R}_i}^{train}$
14:             Compute adapted parameters:
                $\theta_t^i = SGD(\theta_t, \nabla_{\theta_t} \mathcal{L}_{\mathcal{R}_i}(f_{\theta_t}), \alpha)$
15:         end for
16:         Update task parameters:
            $\theta_t = \theta_t - \epsilon \frac{1}{N} \sum_{i=1}^{N} (\theta_t^i - \theta_t)$
17:     until Convergence
18:     Add exemplars of $\mathcal{T}_t$ to $B$
19:     Update $\theta \leftarrow \theta_t$
20: end while
```

6.3 Relation Classification Model

In principle the learner model f_θ could be any gradient-optimized relation classification model. However, in order to use the same number of parameters

Method	*FewRel*		*SimpleQuestions*	
	$ACC_{w.}$	$ACC_{a.}$	$ACC_{w.}$	$ACC_{a.}$
Origin	0.189	0.208	0.632	0.569
GEM	0.492	0.598	0.841	0.796
AGEM	0.361	0.425	0.776	0.722
EWC	0.271	0.302	0.672	0.590
EA-EMR (Full)	0.566	0.673	0.878	0.824
EA-EMR (w/o Sel.)	0.564	0.674	0.857	0.812
EA-EMR (w/o Align.)	0.526	0.632	0.869	0.820
EMR	0.510	0.620	0.852	0.808
MLLRE	**0.602**	**0.741**	**0.880**	**0.842**

Table 6.1 Accuracy on the test set of all tasks ACC_{whole} (denoted $ACC_{w.}$) and average accuracy on the test set of only observed tasks ACC_{avg} (denoted $ACC_{a.}$) on the *Lifelong FewRel* and *Lifelong SimpleQuestions* datasets. Best results are in bold. Except for *MLLRE*, results for other models are obtained from Wang et al. (2019).

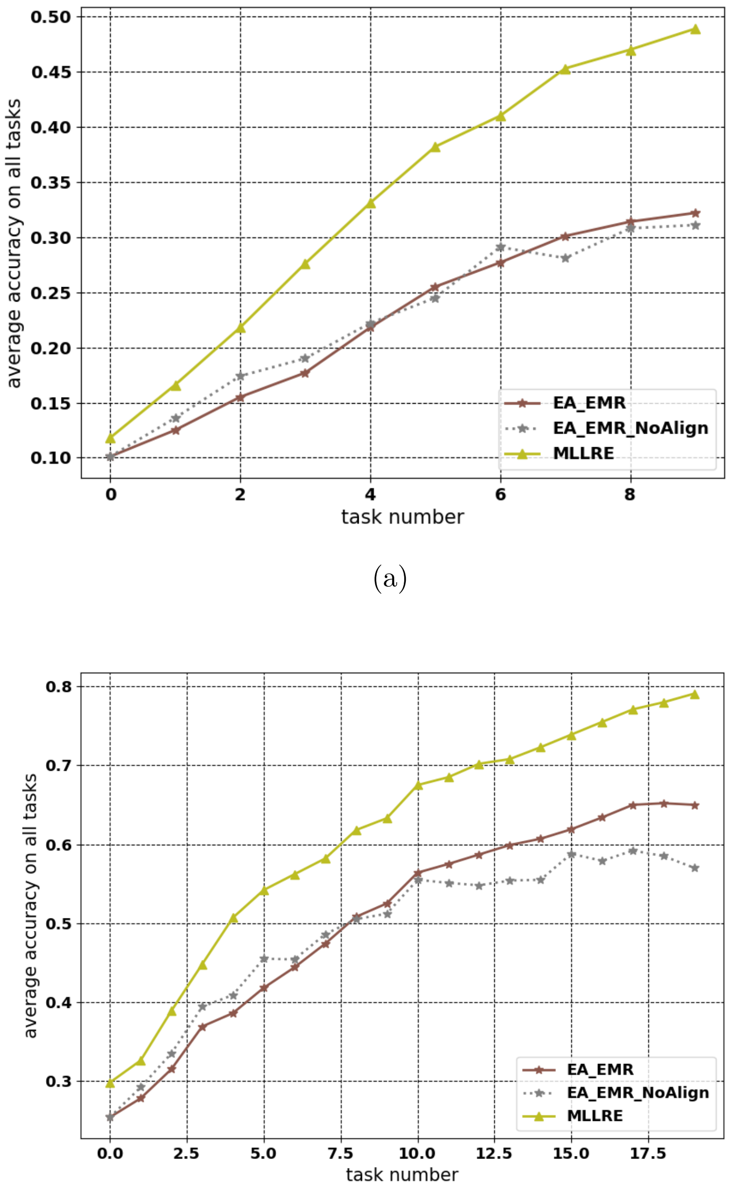
and ensure fair comparison to Wang et al. (2019), we adopt as the relation classification model f_θ the Hierarachical Residual BiLSTM (*HR-BiLSTM*) model of Yu et al. (2017), which is the same model used by Wang et al. (2019) for their experiments. The *HR-BILSTM* is a relation classifier which accepts as input a sentence and a candidate relation, then utilizes two Bidirectional Long Short-Term Memory (Hochreiter and Schmidhuber, 1997; Graves and Schmidhuber, 2005) (BiLSTM) units with shared parameters to process the Glove (Pennington et al., 2014) embeddings of words in the sentence and relation name, and the final representation for each sequence is obtained via max-pooling the BiLSTM outputs of its component words. The model then selects the relation whose representation has maximum cosine similarity to that of the sentence as its response.

Given the BiLSTM representation of a sampled relation $\mathbf{r}^{+}_{rel}$, its (sentence) instance $\mathbf{s}_{sent}$, and a randomly sampled negative relation $\mathbf{r}^{-}_{rel}$, the model is trained with the following ranking loss (Yu et al., 2017):

$$\mathcal{L} = \max\left\{0, \gamma - cosine\left(\mathbf{r}^{+}_{rel}; \mathbf{s}_{sent}\right) + cosine\left(\mathbf{r}^{-}_{rel}; \mathbf{s}_{sent}\right)\right\} \tag{6.2}$$

where *cosine* is the cosine similarity function and γ is a hyperparameter.

Hyperparameters Apart from the hyperparameters specific to meta-learning (such as the step size ϵ), all other hyperparameters we use for the learner model are the same as used by Wang et al. (2019). We also use the

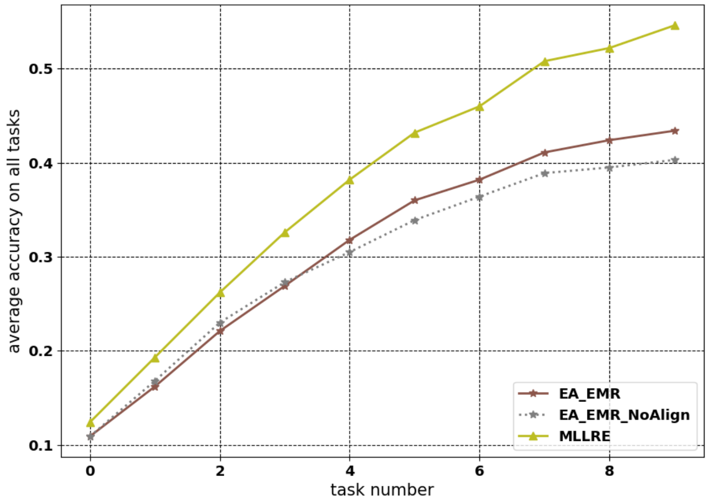


(a)

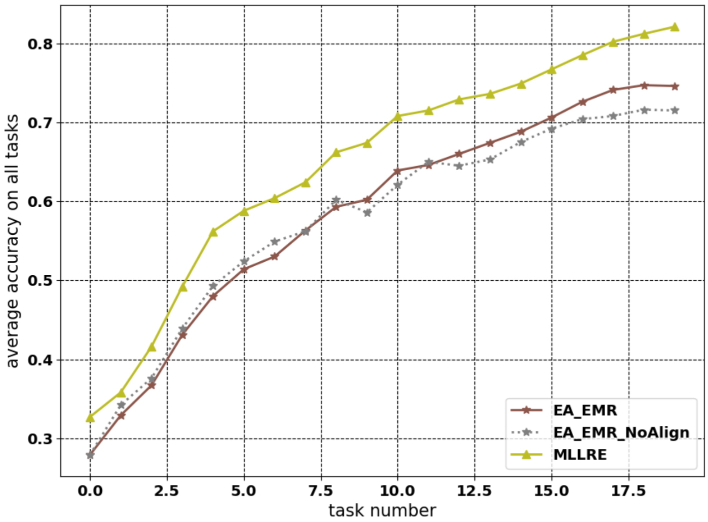
(b)

Figure 6.1 Results obtained using 100 training instances for each task on (a) *Lifelong FewRel* and (b) *Lifelong SimpleQuestions* datasets.

same buffer memory size (50) for each task. Note that the meta-learning algorithm uses SGD as the update rule ($\mathcal{U}$), and does not add any additional trainable parameters to the learner model.



(a)



(b)

Figure 6.2 Results obtained using 200 training instances for each task on (a) *Lifelong FewRel* and (b) *Lifelong SimpleQuestions* datasets.

6.4 Experiments

6.4.1 Setup

We conduct experiments in two settings. In the full supervision setting, we provide all models with all supervision available in the training set of each task. In the second, we limit the amount of supervision for each task to measure how the models are able to cope with limited supervision. Each experiment is run five (5) times and we report the average result.

6.4.2 Datasets

We conduct experiments on *Lifelong FewRel* and *Lifelong SimpleQuestions* datasets, both introduced in Wang et al. (2019). *Lifelong FewRel* is derived from the *FewRel* (Han et al., 2018) dataset, by partitioning its 80 relations into 10 distinct clusters made up of 8 relations each, with each cluster serving as a task where a sentence must be labeled with the correct relation. The 8 relations in each cluster were obtained by clustering the averaged Glove word embeddings of the relation names in the *FewRel* dataset. Each instance of the dataset contains a sentence, the relation it expresses and a set of randomly sampled negative relations. *Lifelong SimpleQuestions* was similarly obtained from the *SimpleQuestions* (Bordes et al., 2015) dataset, and is made up of 20 clusters of relations, with each cluster serving as a task.

6.4.3 Evaluation Metrics

We report two measures, ACC_{whole} and ACC_{avg}, both introduced in Wang et al. (2019). ACC_{whole} measures accuracy on the test set of all tasks and gives a balanced measure of model performance on both observed (seen) and unobserved (unseen) tasks, and is the primary metric we report for all experiments. We also report ACC_{avg}, which measures the average accuracy on the test set of only observed (seen) tasks.

6.4.4 Results and Discussion

Full Supervision Results

Table 6.1 gives both the ACC_{whole} and ACC_{avg} results of our approach compared to other approaches including Episodic Memory Replay (EMR) and its various embedding-aligned variants *EA-EMR* as proposed in Wang et al.

(2019). Across all metrics, our approach outperforms the previous approaches, demonstrating its effectiveness in this setting. This result is likely because our approach is able to efficiently learn new relations by exploiting knowledge from previously observed relations.

Limited Supervision Results

The aim of our limited supervision experiments is to compare the use of an alignment module as proposed by Wang et al. (2019) to using our approach when only limited supervision is available for all tasks. We compare three approaches, Full *EA-EMR* (which uses their alignment module), its variant without the alignment module (*EA-EMR_NoAlign*) and our approach (*MLLRE*). Figures 6.1a and 6.1b show results obtained using 100 supervision instances for each task on *Lifelong FewRel* and *Lifelong SimpleQuestions*. Figures 6.2a and 6.2b show the corresponding plots using 200 supervision instances for each task. From the figures, we observe that the use of a separate alignment model results in only minor gains when supervision for the tasks is limited, whereas the use of our approach leads to wide gains on both datasets.

In summary, because our approach explicitly encourages the model to learn to share and transfer knowledge between relations (by means of the meta-learning objective), the model is able to learn to exploit common structures across relations in different tasks to efficiently learn new relations over time. This leads to the performance improvements obtained by our approach.

6.5 Summary

This chapter investigated the effectiveness of utilizing a gradient-based meta-learning algorithm within a lifelong learning setting to enable relation classification models that are able to learn continually. We show the effectiveness of this approach, both when provided full supervision for new tasks and when provided limited supervision for new tasks, and demonstrated that the proposed approach outperformed current state-of-the-art approaches.

Chapter 7

Conclusions and Future Directions

As stated in Chapter 1, the main aim of this work is to investigate novel approaches for relation classification that does not require extensive amounts of supervision data for relations. Specifically, we set out to address the following research questions:

- Given a text corpus, and limited supervision for relations, how do we utilize existing supervision signals within the corpus, such as the interactions between entities as indicated by their contextual surface patterns, to determine what relations exist between the entities?
- Given just the textual descriptions of relations, is it possible to extract such relations between entities, even without access to any annotated training instances for such relations?
- How do we make existing supervised neural relation classification models, which typically require a lot of annotated training data for each relation, more data efficient?
- How can we develop relation classification models that are able to continually incorporate limited supervision data, as it becomes available, in order to be able to both extract new relations and improve their performance on existing relations?

In order to address these questions, this thesis has proposed both model-specific and model-agnostic methods for relation classification in settings with zero and limited annotated data, including when such data is made available continually. This rest of this chapter provides a summary of our contributions and concludes with directions for future work.

7.1 Research Contributions and Findings

7.1.1 Contextual Pattern Embeddings for Relation Classification

In Chapter 3 we investigated how to make the most of limited available supervision for relations, by leveraging it together with the contextual surface patterns between entities, for effective relation classification. The advantage offered by the use of contextual surface patterns for relation classification is that, unlike the use of propositional logic rules, they are easily available for any domain of interest. In order to model the interactions between contextual surface patterns and relations, we learn embeddings for them within a Factorization Machines (FM) model. Using a FM model for this purpose enabled us to model all possible pairwise interactions between contextual surface patterns and relations with factorized parameters, which is especially helpful for mitigating the sparsity of contextual patterns. Making use of factorized parameters also allows the model to generalize to unobserved interactions between contextual patterns and relations, as well as incorporate any available supervision for relations.

In our experiments, we made use of the NYT dataset used by both Rocktäschel et al. (2015) and Demeester et al. (2016). Both previous works proposed the use of logic rules to provide additional supervision when annotated training data is limited. The propositional rules are used to provide supervision for relations with no training instances. The assumption is that the logic rules are available or can be automatically mined from relevant external resources. This assumption however can be violated in practice, as relevant external resources are not always available, especially in low-resource domains and languages.

When compared to approaches which make use of propositional logic rules as for relation classification, we find that our approach performs better. This is even more telling given that our approach uses less data compared to the previous approaches. In addition, our approach also performs competitively when full supervision is available for relations. This outcome demonstrates that explicitly modelling contextual surface patterns can be beneficial for relation classification, both in a setting where there is a limited supervision instances for relations and even when full supervision is available.

7.1.2 Relation Classification as Textual Entailment

Given no annotated training data for relations, in Chapter 4 we investigated the possibility of relation classification between entities using just the textual descriptions of the relations. In order to achieve this, we proposed formulating the task of relation classification as that of textual entailment, with the sentence containing the entities as premise and the relation description as hypothesis. The task of textual entailment is a well-studied sub-field of natural language processing, and a lot of resources, such as annotated datasets and models, have been developed and made available for it over the years. Formulating relation classification as textual entailment has advantages for the task of relation classification.

Formulating the task of relation classification as textual entailment enables us to utilize existing textual entailment datasets to provide supervision for classifying relations with no labelled data. For instance, in experiments using the MultiNLI textual entailment dataset, we found that just training on the entailment dataset alone, without any additional annotated data for relations, is sufficient to enable us perform zero-shot relation classification. This performance is further improved when the entailment dataset is combined with annotated training data for the relations.

The approach also enables us to utilize existing textual entailment models for the task of relation classification. Though initial work on entailment recognition was done on small annotated datasets, the introduction of the SNLI and MultiNLI datasets, which are fully annotated and orders of magnitude larger than previous entailment datasets, enabled the development of various elaborate neural models for entailment recognition. By formulating relation classification as textual entailment, we were able to utilize these models for the task of classifying relations.

Overall, our experiments demonstrate that classifying relations with textual entailment resources, for instance datasets and models, is a viable approach.

7.1.3 Model-Agnostic Meta-Learning for Relation Classification

Inspired by model-agnostic meta-learning approaches, Chapter 5 proposes a model-agnostic protocol for training supervised relation classification models in limited supervision settings. The algorithm considers each relation as a supervised task, and utilizes a gradient-based meta-learning procedure to learn a parameter initialization to enable relation classifiers learn with limited supervision.

In our experiments, which were conducted using two relation classification models and on two relation classification datasets, we show that our approach markedly improves the performance of the two relation classification models when training data is limited. Further, we also find that when provided full supervision during training, performance is not adversely affected, demonstrating that this approach is effective for relation classification.

7.1.4 Lifelong Relation Classification with Meta-Learning

Chapter 6 developed a meta-learning approach for the task of lifelong relation classification. We proposed to embed meta-learning into the lifelong learning setting of relation classification, to enable relation classification models that are not only able to learn continually without forgetting, but also utilize data efficiently in the process. Experiments conducted on two lifelong relation classification benchmarks showed that our approach leads to improved data-efficiency and performance for lifelong relation classification.

This work demonstrated that it is possible for relation classification models to continually learn and incorporate new supervision data as it becomes available, without any need for substantial retraining, and in a data-efficient manner.

7.2 Future Directions

In this section we discuss the various ways in which the work reported in this thesis can be further extended in the future.

7.2.1 Active Learning for Sample Selection

In our limited supervision experiments, the supervision instances used for training were selected randomly. Though this was done to ensure the general applicability of our proposed approaches, it is likely sub-optimal in terms of performance, as the randomly selected samples are not guaranteed to be able to provide quality learning signals to the model. A better alternative to random sampling of the instances would be the use of various active learning criteria (Settles, 2010) for sample selection, as this would further boost the performance of the various approaches proposed in this thesis.

7.2.2 Multitask Learning Across the KBP Pipeline

We have focused specifically on the task of relation classification in this work. However, relation classification is a just one component in the overall knowledge base population pipeline (Ji and Grishman, 2011). Other important tasks, which include entity recognition, coreference resolution and entity linking were not explored in this thesis. It is likely that by exploiting learning signals from these tasks jointly, we can further improve the data efficiency of our approaches.

7.2.3 Multilingual and Multidomain Relation Extraction

A promising direction of exploration for the methods proposed in this thesis is their application to languages other than English, and other domains, for instance documents from the biomedical domain. It is possible that there are data efficiency gains that can be obtained by leveraging data from multiple languages and domains.

Bibliography

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., and Others (2016). TensorFlow: A System for Large-Scale Machine Learning. In *OSDI*, volume 16, pages 265–283.

Adel, H., Roth, B., and Schütze, H. (2016). Comparing convolutional neural networks to traditional models for slot filling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 828–838, San Diego, California. Association for Computational Linguistics.

Agichtein, E. and Gravano, L. (2000). Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94. ACM.

Alfonseca, E., Filippova, K., Delort, J.-Y., and Garrido, G. (2012). Pattern learning for relation extraction with a hierarchical topic model. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 54–59.

Andrychowicz, M., Denil, M., Colmenarejo, S. G., Hoffman, M. W., Pfau, D., Schaul, T., Shillingford, B., and de Freitas, N. (2016). Learning to learn by gradient descent by gradient descent. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 3988–3996, USA. Curran Associates Inc.

Angeli, G., Tibshirani, J., Wu, J. Y., and Manning, C. D. (2014). Combining Distant and Partial Supervision for Relation Extraction. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP2014)*, pages 1556–1567.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). DBpedia: A nucleus for a Web of open data. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4825 LNCS:722–735.

Augenstein, I., Rocktäschel, T., Vlachos, A., and Bontcheva, K. (2016). Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885, Austin, Texas. Association for Computational Linguistics.

Bach, N. and Badaskar, S. (2007). A review of relation extraction. *Literature review for Language and Statistics II.*

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007). Open information extraction from the web. In *Ijcai*, volume 7, pages 2670–2676.

Bengio, Y., Simard, P., Frasconi, P., and Others (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008a). Freebase: a collaboratively created graph database for structuring human knowledge. *SIGMOD 08 Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008b). Freebase: a collaboratively created graph database for structuring human knowledge. *SIGMOD 08 Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Bordes, A., Usunier, N., Chopra, S., and Weston, J. (2015). Large-scale Simple Question Answering with Memory Networks.

Bos, J. and Markert, K. (2005). Recognising textual entailment with logical inference. *Human Language Technology Conference*, (October):628.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *EMNLP*.

Bowman, S. R., Gauthier, J., Rastogi, A., Gupta, R., Manning, C. D., and Potts, C. (2016). A Fast Unified Model for Parsing and Sentence Understanding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1466–1477, Berlin, Germany. Association for Computational Linguistics.

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

Bunescu, R. and Mooney, R. J. (2005a). A shortest path dependency kernel for relation extraction. *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Vancouver, B.C*, (October).

Bunescu, R. and Mooney, R. J. (2005b). Subsequence Kernels for Relation Extraction. *Submitted to the Ninth Conference on Natural Language Learning*, 18.

Bunescu, R. C. and Mooney, R. J. (2007). Learning to Extract Relations from the Web using Minimal Supervision. *Computational Linguistics*, 45(June):576–583.

Carlson, A., Betteridge, J., and Kisiel, B. (2010). Toward an Architecture for Never-Ending Language Learning. *In Proceedings of the Conference on Artificial Intelligence (AAAI) (2010)*, pages 1306–1313.

Chang, M.-W., Ratinov, L.-A., Rizzolo, N., and Roth, D. (2008). Learning and Inference with Constraints. In *AAAI*, pages 1513–1518.

Chaudhry, A., Ranzato, M., Rohrbach, M., and Elhoseiny, M. (2018). Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*.

Chen, D., Bolton, J., and Manning, C. D. (2016a). A Thorough Examination of the CNN / Daily Mail Reading Comprehension Task. *Acl 2016*, pages 2358–2367.

Chen, Q., Zhu, X., Ling, Z., Wei, S., Jiang, H., and Inkpen, D. (2016b). Enhanced LSTM for Natural Language Inference. (2008).

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Collins, M. (2002). Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 1–8. Association for Computational Linguistics.

Collins, M. and Duffy, N. (2002). Convolution kernels for natural language. In *Advances in neural information processing systems*, pages 625–632.

Condoravdi, C., Crouch, D., De Paiva, V., Stolle, R., and Bobrow, D. G. (2003). Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 workshop on Text meaning*, pages 38–45.

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

Craven, M. and Kumlien, J. (1999). Constructing biological knowledge bases by extracting information from text sources. *Proceedings of the International Conference on Intelligent Systems for Molecular Biology (ISMB-99)*, pages 77–86.

Culotta, A., Culotta, A., McCallum, A., McCallum, A., Betz, J., and Betz, J. (2006). Integrating Probabilistic Extraction Models and Data Mining to Discover Relations and Patterns in Text. *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, (June):296–303.

Culotta, A. and Sorensen, J. (2004). Dependency tree kernels for relation extraction. *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics - ACL '04*, 4:423.

Curran, J. R., Murphy, T., and Scholz, B. (2007). Minimising semantic drift with Mutual Exclusion Bootstrapping.

Dagan, I., Glickman, O., and Magnini, B. (2005). The PASCAL Recognising Textual Entailment Challenge. In *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, pages 177–190.

De Lacalle, O. L. and Lapata, M. (2013). Unsupervised relation extraction with general domain knowledge. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 415–425.

Demeester, T., Rocktäschel, T., and Riedel, S. (2016). Lifted Rule Injection for Relation Embeddings. *Acl*.

Doddington, G. R., Mitchell, A., Przybocki, M. A., Ramshaw, L. A., Strassel, S. M., and Weischedel, R. M. (2004). The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation. In *Lrec*, volume 2, page 1. Lisbon.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211.

Fader, A., Soderland, S., and Etzioni, O. (2011). 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.

Fernando, C., Banarse, D., Blundell, C., Zwols, Y., Ha, D., Rusu, A. A., Pritzel, A., and Wierstra, D. (2017). Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks.

Finn, C., Rajeswaran, A., Kakade, S., and Levine, S. (2019). Online Meta-Learning.

Firth, J. R. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.

French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Fyodorov, Y., Winter, Y., and Francez, N. (2000). A natural logic inference system. In *Proceedings of the 2nd Workshop on Inference in Computational Semantics (ICoS-2)*. Citeseer.

Gantz, J., Reinsel, D., and Shadows, B. D. (2012). The Digital Universe in 2020. *IDC iView "Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East"*, 2007(December 2012):1–16.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741.

Glickman, O., Dagan, I., and Koppel, M. (2006). A lexical alignment model for probabilistic textual entailment. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3944 LNAI, pages 287–298.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Graves, A., Jaitly, N., and Mohamed, A.-r. (2013). Hybrid speech recognition with deep bidirectional LSTM. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 273–278. IEEE.

Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM networks. In *Proceedings of the International Joint Conference on Neural Networks*, volume 4, pages 2047–2052.

Grishman, Ralph Sundheim, B. (1996). Message Understanding Conference-6: A Brief History. In *{COLING} 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Haghighi, A. and Vanderwende, L. (2009). Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370. Association for Computational Linguistics.

Han, X., Zhu, H., Yu, P., Wang, Z., Sun, M., Yao, Y., and Liu, Z. (2018). FewRel : A Large-Scale Supervised Few-Shot Relation Classification Dataset with State-of-the-Art Evaluation. In *Emnlp*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

Hasegawa, T., Sekine, S., and Grishman, R. (2004). Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics - ACL '04*, pages 415–es.

Hashimoto, K., Miwa, M., Tsuruoka, Y., and Chikayama, T. (2013). Simple Customization of Recursive Neural Networks for Semantic Relation Classification. In *Proceedings of the 2013 Conference on Empirical Methods in*

Natural Language Processing, pages 1372–1376. Association for Computational Linguistics.

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L., and Szpakowicz, S. (2010). SemEval-2010 Task 8 : Multi-Way Classification of Semantic Relations Between Pairs of Nominals. *Computational Linguistics*, (June 2009):94–99.

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Séaghdha, D. O., Padó, S., Pennacchiotti, M., Romano, L., and Szpakowicz, S. (2009). SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations Between Pairs of Nominals. *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, (July):94–99.

Hewlett, D., Lacoste, A., Jones, L., Polosukhin, I., Fandrianto, A., Han, J., Kelcey, M., and Berthelot, D. (2016). WikiReading: A novel large-scale language understanding task over Wikipedia. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1545, Berlin, Germany. Association for Computational Linguistics.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., and Weld, D. S. (2011). 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*, pages 541–550, Portland, Oregon, USA. Association for Computational Linguistics.

Jensen, C. S., Kjærulff, U., and Kong, A. (1995). Blocking Gibbs sampling in very large probabilistic expert systems. *International Journal of Human-Computer Studies*, 42(6):647–666.

Ji, H. and Grishman, R. (2011). Knowledge Base Population : Successful Approaches and Challenges. *Acl*, pages 1148–1158.

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

Jiang, J. and Zhai, C. (2007). A systematic exploration of the feature space for relation extraction. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 113–120.

Jijkoun, V. and de Rijke, M. (2005). Recognizing Textual Entailment using Lexical Similarity. *Recognizing Textual Entailment*, page 73.

Jordan, M. I. (1986). Serial order: A parallel distributed processing approach, ICS Report 8604. *Institute for Cognitive Science, UCSD, La Jolla.*

Kambhatla, N. (2004). Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 22.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv:1412.6980 [cs.LG]*, pages 1–13.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., and Others (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Konstantinova, N. (2014). Review of relation extraction methods: What is new out there? In *International Conference on Analysis of Images, Social Networks and Texts*, pages 15–28. Springer.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., and Others (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Levy, O., Seo, M., Choi, E., and Zettlemoyer, L. (2017). Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Lewis, M. and Steedman, M. (2013). Unsupervised induction of cross-lingual semantic relations. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 681–692.

Lin, D. and Pantel, P. (2001). DIRT - discovery of inference rules from text. *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining KDD 01*, datamining:323–328.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. (2002). Text classification using string kernels. *Journal of Machine Learning Research*, 2(Feb):419–444.

Lopez-Paz, David and Ranzato, M. (2017). Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476.

MacCartney, B. and Manning, C. D. (2009). An extended model of natural logic. In *Proceedings of the eighth international conference on computational semantics*, pages 140–156. Association for Computational Linguistics.

MacKay, D. J. C. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

McCann, B., Bradbury, J., Xiong, C., and Socher, R. (2017). Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6297–6308.

McClelland, J. L., McNaughton, B. L., and O'reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419.

McCloskey, M. and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Min, B., Grishman, R., Wan, L., Wang, C., and Gondek, D. (2013). Distant Supervision for Relation Extraction with an Incomplete Knowledge Base. In *Proceedings of the 2013 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 777–782, Atlanta, Georgia. Association for Computational Linguistics.

Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, (August):1003–1011.

Muslea, I., Minton, S., and Knoblock, C. A. (2000). Selective sampling with redundant views. In *AAAI/IAAI*, pages 621–626.

Naik, D. K. and Mammone, R. J. (1992). Meta-neural networks that learn by learning. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, volume 1, pages 437–442. IEEE.

Nemirovski, A. and Yudin, D. (1978). On Cezari's convergence of the steepest descent method for approximating saddle point of convex-concave functions. In *Soviet Math. Dokl*, volume 19, pages 258–269.

Nguyen, D. P. T., Matsuo, Y., and Ishizuka, M. (2007). Exploiting syntactic and semantic information for relation extraction from wikipedia. In *IJCAI Workshop on Text-Mining & Link-Analysis (TextLink 2007)*.

Nguyen, T. H. and Grishman, R. (2015). Relation Extraction: Perspective from Convolutional Neural Networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48, Denver, Colorado. Association for Computational Linguistics.

Nguyen, T.-V. T. and Moschitti, A. (2011). Joint distant and direct supervision for relation extraction. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 732–740.

Nguyen, T.-v. T., Moschitti, A., and Riccardi, G. (2009a). 2009Convolution kernels on constituent, dependency and sequential structures for relation extraction.pdf. (August):1378–1387.

Nguyen, T.-V. T., Moschitti, A., and Riccardi, G. (2009b). Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1378–1387. Association for Computational Linguistics.

Nichol, A., Achiam, J., and Schulman, J. (2018). On First-Order Meta-Learning Algorithms. *arXiv preprint*.

Parikh, A. P., Täckström, O., Das, D., and Uszkoreit, J. (2016). A Decomposable Attention Model for Natural Language Inference.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318.

Pawar, S., Palshikar, G. K., and Bhattacharyya, P. (2017). Relation Extraction : A Survey. pages 1–51.

Pennington, J., Socher, R., and Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Pershina, M., Min, B., Xu, W., and Grishman, R. (2014). Infusion of labeled data into distant supervision for relation extraction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 732–738.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proc. of NAACL*.

Petroni, F., Del Corro, L., and Gemulla, R. (2015). CORE: Context-Aware Open Relation Extraction with Factorization Machines. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (September):1763–1773.

Ratcliff, R. (1990). Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285.

Ravi, S. and Larochelle, H. (2017). Optimization As a Model for Few-Shot Learning. In *International Conference on Learning Representations 2017*.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. (2017). icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2001–2010.

Rendle, S. (2010). Factorization machines. *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 995–1000.

Rendle, S. (2012). Factorization Machines with libFM. *ACM Trans. Intell. Syst. Technol. Article*, 3(22).

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009a). BPR: Bayesian Personalized Ranking from Implicit Feedback. *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461.

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009b). BPR: Bayesian Personalized Ranking from Implicit Feedback. *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461.

Riedel, S., Yao, L., and McCallum, A. (2010). Modeling relations and their mentions without labeled text. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 6323 LNAI, pages 148–163.

Riedel, S., Yao, L., McCallum, A., and Marlin, B. M. (2013). Relation Extraction with Matrix Factorization and Universal Schemas. *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (June):74–84.

Riloff, E. and Jones, R. (1999). Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. In *AAAI/IAAI*, pages 474–479.

Ring, M. B. (1994). *Continual learning in reinforcement environments.* PhD thesis, University of Texas at Austin Austin, Texas 78712.

Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kocisky, T., and Blunsom, P. (2016). Reasoning about Entailment with Neural Attention. In *Proceedings of the International Conference on Learning Representations (ICLR).*

Rocktäschel, T., Singh, S., and Riedel, S. (2015). Injecting Logical Background Knowledge into Embeddings for Relation Extraction. *North American Association for Computational Linguistics*, pages 1119–1129.

Romano, L., Kouylekov, M., Szpektor, I., Dagan, I., and Lavelli, A. (2006). Investigating a generic paraphrase-based approach for relation extraction. In *11th Conference of the European Chapter of the Association for Computational Linguistics.*

Roth, B., Barth, T., Wiegand, M., and Klakow, D. (2013). A survey of noise reduction methods for distant supervision. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 73–78. ACM.

Roth, D., Sammons, M., and Vydiswaran, V. G. V. (2009). A Framework for Entailed Relation Recognition. *Ijcnlp2009*, (August):57–60.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., and Others (1988). Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. (2016). Progressive neural networks. *arXiv preprint arXiv:1606.04671.*

Sandhaus, E. (2008). The New York Times Annotated Corpus.

Sarawagi, S. (2008). Information extraction. *Foundations and Trends®in Databases*, 1(3):261–377.

Schmidhuber, J. (1987). Evolutionary principles in self-referential learning. *On learning how to learn: The meta-meta-... hook.) Diploma thesis, Institut f. Informatik, Tech. Univ. Munich.*

Settles, B. (2010). Active Learning Literature Survey.

Snell, J., Swersky, K., and Zemel, R. S. (2017). Prototypical Networks for Few-shot Learning.

Socher, R., Huval, B., Manning, C. D., and Ng, A. Y. (2012). Semantic Compositionality through Recursive Matrix-Vector Spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211. Association for Computational Linguistics.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). YAGO: A Core of Semantic Knowledge Unifying Wordnet and Wikipedia. *Proceedings of the 16th international conference on World Wide Web*, pages 697–706.

Sun, A. and Grishman, R. (2012). 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.

Surdeanu, M., Tibshirani, J., Nallapati, R., and Manning, C. D. (2012a). Multi-instance Multi-label Learning for Relation Extraction. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP '12*, (July):455–465.

Surdeanu, M., Tibshirani, J., Nallapati, R., and Manning, C. D. (2012b). Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465, Jeju Island, Korea. Association for Computational Linguistics.

Takamatsu, S., Sato, I., and Nakagawa, H. (2012). Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 721–729. Association for Computational Linguistics.

Thrun, S. (1996). Is learning the n-th thing any easier than learning the first? In *Advances in neural information processing systems*, pages 640–646.

Thrun, S. and Pratt, L. (1998). Learning to Learn: Introduction and Overview. In *Learning to Learn*, pages 3–17. Springer US, Boston, MA.

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., and Wierstra, D. (2016). Matching Networks for One Shot Learning.

Vrandečić, D. (2012). Wikidata: a new platform for collaborative data collection. *Proceedings of the 21st international conference companion on World Wide Web - WWW '12 Companion*, page 1063.

Vu, N. T., Adel, H., Gupta, P., and Schütze, H. (2016). Combining Recurrent and Convolutional Neural Networks for Relation Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 534–539. Association for Computational Linguistics.

Wang, H., Xiong, W., Yu, M., Guo, X., Chang, S., and Wang, W. Y. (2019). Sentence embedding alignment for lifelong 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 796–806, Minneapolis, Minnesota. Association for Computational Linguistics.

Welbl, J., Bouchard, G., and Riedel, S. (2016). A Factorization Machine Framework for Testing Bigram Embeddings in Knowledgebase Completion. *ArXiv*, pages 103–107.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

Wick, M., Rohanimanesh, K., Culotta, A., and McCallum, A. (2009). Samplerank: Learning preferences from atomic gradients. In *Neural Information Processing Systems (NIPS), Workshop on Advances in Ranking*.

Williams, A., Nangia, N., and Bowman, S. R. (2017). A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference.

Xiao, T., Zhang, J., Yang, K., Peng, Y., and Zhang, Z. (2014). Error-driven incremental learning in deep convolutional neural network for large-scale image classification. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 177–186. ACM.

Xu, W., Grishman, R., and Zhao, L. (2011). Passage retrieval for information extraction using distant supervision. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1046–1054.

Xu, W., Hoffman, R., Zhao, L., and Grishman, R. (2013). Filling Knowledge Base Gaps for Distant Supervision of Relation Extraction. *ACL*, pages 665–670.

Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H., and Jin, Z. (2015). Classifying Relations via Long Short Term Memory Networks along Shortest Dependency Paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1785–1794, Lisbon, Portugal. Association for Computational Linguistics.

Yao, L., Riedel, S., and Mccallum, A. (2011). Structured Relation Discovery using Generative Models. *Technology*, pages 1456–1466.

Yu, M., Yin, W., Hasan, K. S., dos Santos, C., Xiang, B., and Zhou, B. (2017). Improved Neural Relation Detection for Knowledge Base Question Answering. pages 571–581.

Zelenko, D., Aone, C., and Richardella, A. (2003). Kernel methods for relation extraction. *The Journal of Machine Learning Research*, 3:1083–1106.

Zeng, D., Liu, K., Lai, S., Zhou, G., and Zhao, J. (2014). Relation Classification via Convolutional Deep Neural Network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Zenke, F., Poole, B., and Ganguli, S. (2017). Continual Learning Through Synaptic Intelligence. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3987–3995, International Convention Centre, Sydney, Australia. PMLR.

Zhang, D. and Wang, D. (2015). Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*.

Zhang, M., Zhang, J., Su, J., and Zhou, G. (2006). 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 the 44th annual meeting of the Association for Computational Linguistics*, pages 825–832. Association for Computational Linguistics.

Zhang, Y., Qi, P., and Manning, C. D. (2018). Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium. Association for Computational Linguistics.

Zhang, Y., Zhong, V., Chen, D., Angeli, G., and Manning, C. D. (2017). Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

Zhao, J. and Schmidhuber, J. (1996). Incremental self-improvement for life-time multi-agent reinforcement learning. In *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior, Cambridge, MA*, pages 516–525.

Zhao, S. and Grishman, R. (2005). Extracting relations with integrated information using kernel methods. *In ACL*, pages 419–426.