Doctoral Dissertation

Relation Extraction: Perspective from Various Supervised Approaches

Tran Van Hien
Program of Information Science and Engineering
Graduate School of Science and Technology
Nara Institute of Science and Technology

Supervisor: Professor Taro Watanabe
Natural Language Processing Lab. (Division of Information Science)

Submitted on September 10, 2022

A Doctoral Dissertation
submitted to Graduate School of Science and Technology,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Tran Van Hien

Thesis Committee:

Professor Taro Watanabe	(Supervisor)
Professor Satoshi Nakamura	(Co-supervisor)
Associate Professor Hiroyuki Shindo	(Co-supervisor)
Assistant Professor Hiroki Ouchi	(Co-supervisor)
Doctor Yuji Matsumoto	(RIKEN AIP)

Relation Extraction: Perspective from Various Supervised Approaches[1]

Tran Van Hien

Abstract

Information extraction transforms unstructured text into structured information on raw data. A vital step in information extraction is relation extraction, which aims to identify semantic relationships between named entities in text. The extracted relations help construct knowledge bases and support various natural language processing applications such as information retrieval and question answering.

Relation extraction has been widely studied in a fully supervised learning approach by training models on large-scale labeled data. Following this approach, existing supervised models have achieved excellent performance. However, these supervised models cannot solve relation extraction in real-world scenarios, such as recognizing new relations or identifying entities and their relations jointly.

In this dissertation, we focus on two other supervised approaches for relation extraction task, namely *zero-shot relation extraction* and *end-to-end relation extraction*. These two supervised approaches help solve relation extraction in real-world scenarios, which are more realistic and challenging.

The first part of this dissertation addresses *zero-shot relation extraction*, which aims to recognize (new) unseen relations that cannot be observed during train-

[1]Doctoral Dissertation, Graduate School of Science and Technology, Nara Institute of Science and Technology, September 10, 2022.

ing. We propose two new methods to improve task performance. In the first method, we present a new model that mainly boosts discriminative feature learning on both sentence and relation spaces. This model is also equipped with a self-adaptive comparator network to judge whether the relationship between a sentence and a relation is consistent. Experimental results show that the proposed method significantly outperforms the state-of-the-art methods. In the second method, we argue that enhancing the semantic correlation between instances and relations is key to solving the zero-shot relation extraction task effectively. A new model entirely devoted to this goal through three main aspects was proposed: learning effective relation representation, designing purposeful mini-batches, and binding two-way semantic consistency. Experimental results on two benchmark datasets demonstrate that our approach significantly improves task performance and achieves state-of-the-art results.

The second part of this study concentrates *end-to-end relation extraction*, which aims to detect entity pairs along with their relations to extract relational triplets. We propose an improved decomposition strategy that overcomes two major problems of the previous decomposition strategy by Yu et al. (2020). Our improved decomposition strategy considers each extracted entity in two roles (*head* and *tail*) and allows a model to predict multiple relations (if any) of an entity pair. In addition, a corresponding model framework is presented to deploy our new decomposition strategy. Experimental results show that our method significantly outperformed the previous method of Yu et al. (2020) and achieved state-of-the-art performance on two benchmark datasets. Besides, we also present CovRelex (Tran et al., 2021), a scientific paper retrieval system that can automatically detect both entities with various types and their diverse relations through papers, primarily when COVID-19 articles are published rapidly. The system aims to support users efficiently in acquiring such knowledge across many COVID-19 scientific papers.

Keywords:

relation extraction, fully supervised learning, zero-shot learning, joint extraction, supervised learning, end-to-end learning, decomposition strategy, covid-19 relation extraction, neural networks

Acknowledgements

For me, the Ph.D. journey is a roller coaster; without the support of many people, I would never have gone this far. In particular, I consider myself lucky to have had great advisors during my graduate life.

First, I would like to express my sincere appreciation and gratitude to Professor Taro Watanabe. He always gave me his best support as soon as possible and encouraged me to become a better researcher. His numerous insightful comments profoundly shaped the work in this dissertation. I wish I could have more time to work with and learn more from him.

Second, I am incredibly grateful to Doctor Yuji Matsumoto, who formerly led the computational linguistics lab at NAIST and was my supervisor for two and a half years. His continuous support and inspiring suggestions have been precious for developing my master thesis in 2019 and this doctoral dissertation. I am thankful to him for being a member of my dissertation committee.

I would also like to thank Professor Satoshi Nakamura, Associate Professor Hiroyuki Shindo, and Assistant Professor Hiroki Ouchi as co-supervisors for their kind instructions and valuable comments on my work. Thanks so much for all the great, thoughtful feedback on my research and this dissertation.

Being a member of the NAIST NLP lab for almost five years has been fun and exciting. I appreciate the uninterrupted discussions, valuable comments, and memorable activities that I shared with colleagues, alumni, faculty, and staff in the group. I am very proud to have worked with them and been a part of the group. I also want to thank Ms. Yuko Kitagawa, my lab assistant, who has always provided significant help in the lab and in daily life.

Last but not least, I would like to thank my family for their unconditional love and tireless support throughout my life and studies.

Thank you very much, everyone!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In the current digital age, people easily create, share, and obtain information on the Internet, leading to the exponential growth of various digital contents such as images, video, speech, and text. It is infeasible for humans to read through such a large amount of text. Thus, we expect computers to automatically understand natural language to extract meaningful information in desirable structures.

Information extraction, an important area of natural language processing, develops methods to support computers for this target. It aims to transform unstructured text into machine-readable structures for further applications such as knowledge base construction, question answering, and information retrieval. In particular, information extraction methods disclose the underlying structures by recognizing entities and semantic relations between them. Such methods help readers grasp essential information over a large amount of text.

In this dissertation, we study **relation extraction**, a sub-field of information extraction. Relation extraction aims to identify semantic relations between named entities within a given unstructured text.

Previous studies considered relation extraction in a fully supervised learning approach, which identifies semantic relation between given pairs of entities by training models on large-scaled labeled datasets. Following this approach, traditional models usually rely on heavily haft-crafted features and linguistic resources, or elaborately designed kernels, which are time-consuming and challenging to adapt to novel domains. Recently, neural network models have dominated this

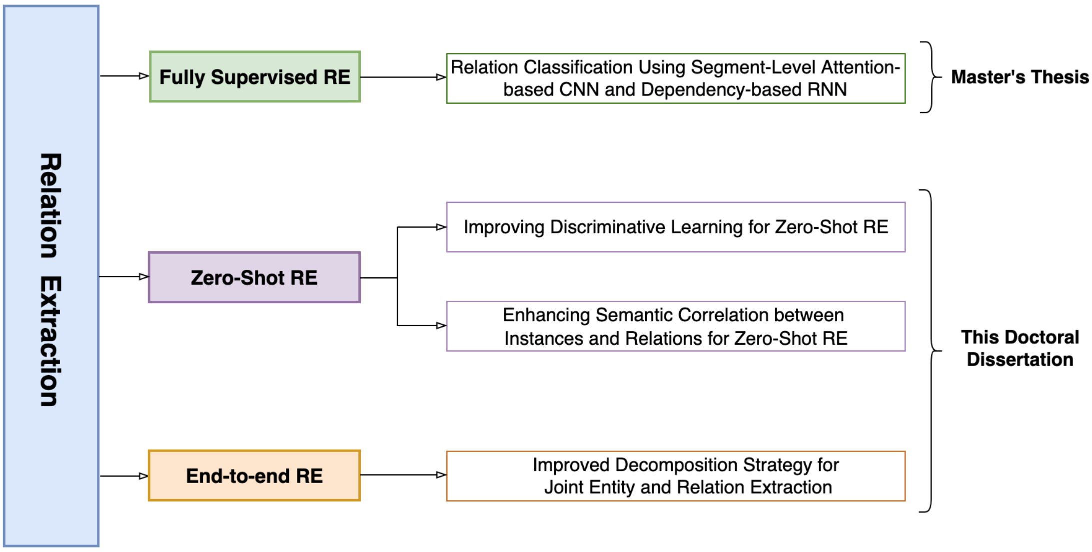

Figure 1.1: A high-level overview of my research.

task since they can effectively learn meaningful hidden features without human intervention. Existing supervised neural network models have achieved excellent performance on this approach. However, these supervised models cannot solve the relation extraction task in real-world scenarios, such as recognizing new relations or identifying entities and their relations jointly. Thus, in this study, instead of only considering relation extraction in a fully supervised classification approach, we further deal with this task in two other supervised approaches: *zero-shot relation extraction* and *end-to-end relation extraction*, where *zero-shot relation extraction* focuses on recognizing new relations and *end-to-end relation extraction* aims to extract entities and their relations jointly.

The high-level overview of my research is shown in Figure 1.1. In my Master's thesis (Tran, 2019), we dealt with relation extraction in a fully supervised learning approach. Specifically, in our work (Tran et al., 2019), we proposed a new model effectively combining Segment-level Attention-based Convolutional Neural Networks (SACNNs) and Dependency-based Recurrent Neural Networks (DepRNNs). While SACNNs allow the model to selectively focus on the vital information segment from the raw sequence, DepRNNs help handle the long-distance relations from the shortest dependency path between the related entities. Experiments on the SemEval-2010 Task 8 dataset showed that our model is comparable to the state-of-the-art without using any external lexical features.

In this doctoral dissertation, we further consider relation extraction in the two other supervised approaches: *zero-shot relation extraction* and *end-to-end relation extraction*, which are more challenging and realistic in real-world scenarios.

First, "*zero-shot relation extraction*" aims to recognize (new) unseen relations that cannot be observed during the training phase. Due to the lack of information, recognizing unseen relations with no corresponding labeled training instances is a challenging task. We propose two new methods to improve task performance. In the first method, we present a new model incorporating discriminative embedding learning for both sentences and semantic relations. In addition, a self-adaptive comparator network is used to judge whether the relationship between a sentence and a relation is consistent. Experimental results on two benchmark datasets show that the proposed method significantly outperforms the state-of-the-art methods. In the second method, we argue that enhancing the semantic correlation between instances and relations is a key to solving the zero-shot relation extraction task effectively. A new model entirely devoted to this goal through three main aspects was proposed: learning effective relation representation, designing purposeful mini-batches, and binding two-way semantic consistency. Experimental results on two benchmark datasets demonstrate that our method significantly improves task performance and achieves state-of-the-art results.

Second, "*end-to-end relation extraction*" is a critical and challenging task in NLP. Given an unstructured text, it aims to extract pairs of entities with semantic relations to create relational triplets, in the form of (*head entity, relation, tail entity*). One of the biggest challenges of this task is the overlapping triplet problem, where the same entity pair exists multiple semantic relations or two different triplets overlap one entity. To alleviate this problem, Yu et al. (2020) presented a novel decomposition strategy that decomposes this task into two interrelated subtasks, namely *head entity extraction* and *tail entity relation extraction*. However, this strategy still has some limitations that hinder the model from solving the problem effectively. We, therefore, propose an improved decomposition strategy that overcomes the existing limitations of the previous strategy. Experimental results show that our method significantly outperformed the method of Yu et al. (2020) and achieved state-of-the-art performance on two benchmark datasets. Furthermore, we exploit *end-to-end relation extraction* in a realistic project to process COVID-19 scientific papers. Due to the COVID-19 outbreak, researchers have been focusing on studying the virus and publishing a large number of COVID-19-related scientific papers rapidly. Thus, it is essential to grasp valuable knowledge from these papers for dealing with the pandemic effectively. We present CovRelex (Tran et al., 2021), a scientific retrieval system that focuses on grasping entities and their relations. Specifically, the CovRelex can automatically detect entities with various types and their diverse relations through papers. By acquiring such valuable knowledge of biomedical entities, CovRelex can answer several questions regarding the entities and their relations with users.

1.2 Contribution

The main contribution of this dissertation are as follows:

- A new model incorporating discriminative embedding learning for both sentences and semantic relations is proposed for zero-shot relation extraction task.

- Experimental results on two benchmark datasets showed that the proposed model significantly outperforms the state-of-the-art methods in the zero-shot relation extraction task.

- A new method that focuses on enhancing this semantic correlation by learning high-quality relation representation, designing strategic mini-batches, and binding two-way semantic consistency is proposed.

- Extensive experiments on two benchmark datasets demonstrated the effectiveness and robustness of the new method, as it significantly outperformed the existing state-of-the-art methods.

- For the end-to-end relation extraction task, an improved decomposition strategy is presented to overcome some limitations of the prior decomposition strategy by Yu et al. (2020).

- A corresponding model framework is introduced to deploy the new decomposition strategy for the end-to-end relation extraction.
- Experimental results showed that the new decomposition strategy significantly outperformed the previous approach of Yu et al. (2020) and achieved state-of-the-art performance on two benchmark datasets.

1.3 Organization of the Dissertation

This dissertation is structured as follows:

- Chapter 1 presents this dissertation's motivation, contributions, and organization.
- Chapter 2 provides a background of relation extraction and related work on various supervised approaches for this task.
- Chapter 3 introduces our proposed method of improving discriminative learning for zero-shot relation extraction.
- Chapter 4 presents our new method that focuses on enhancing semantic correlation between instances and relations for solving zero-shot relation extraction.
- Chapter 5 investigates the effectiveness of our improved decomposition strategy for joint entity and relation extraction.
- Chapter 6 concludes the dissertation with a summary of research results, open problems, and future work for the relation extraction task.

Chapter 2

Background and Related Work

As introduced in Figure 1.1, the overview of my research investigates relation extraction task into three different supervised approaches: *fully supervised relation extraction*, *zero-shot relation extraction*, and *end-to-end relation extraction*. First, we introduce background on relation extraction and each of the three supervised approaches for this task in detail. Then, we present related work on *zero-shot relation extraction* and *end-to-end relation extraction* since this dissertation focuses on these two supervised approaches.

2.1 Background

2.1.1 Relation Extraction Task

Relation extraction is a fundamental task in natural language processing (NLP) that aims to recognize semantic relations between concepts, also called named entities or arguments. A *named entity*, known also as *entity*, can be expressed by a word or a sequence of words that indicate a concept of interest. Figure 2.1 illustrates a semantic relation between two entities: *Edsel Ford* and *Henry Ford* in a given sentence[1].

Relation extraction (RE) has attracted much research effort as it plays a vital role in many NLP applications. Specifically, the extracted results can be used in downstream applications such as information retrieval (Wei et al., 2013; Soto et al., 2019), textual entailment (Szpektor et al., 2004; Eichler et al., 2016), and

[1]In this dissertation, "sentence" and "instance" are interchangeable.

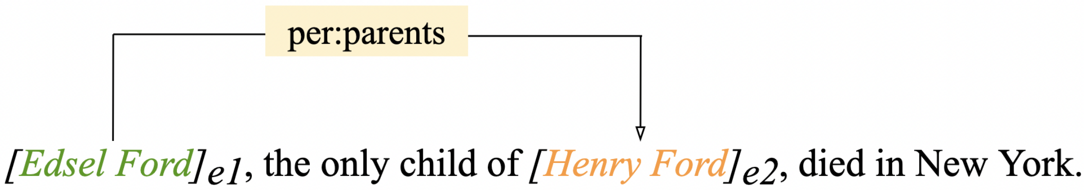

Figure 2.1: An example expressing the semantic relation between two entities in a given sentence from the TACRED dataset (Zhang et al., 2017).

question answering (Xu et al., 2016). Entities that participate in a relation can be located within a sentence, in a short paragraph, or in a document. Previous work mainly studies sentence-level relation extraction (intra-sentence RE). In the scope of this dissertation, we also focus on identifying semantic relations between entities within a single sentence. However, in reality, many entities can have semantic relations across sentences (inter-sentence), either in a paragraph or a document. Recognizing relations between entities over multiple sentences will be our future work.

2.1.2 Fully Supervised Relation Extraction

Traditionally, a fully supervised relation extraction task is naturally cast as a supervised classification problem. Conventional approaches (Kambhatla, 2004a; Zhang et al., 2006b; Chan and Roth, 2010; Sun et al., 2011; Nguyen and Grishman, 2014; Nguyen et al., 2015) usually rely heavily on linguistic and hand-crafted features, or elaborately designed kernels, which are time-consuming and challenging to adapt to new domains. Recently, neural network models have dominated the work on fully supervised relation extraction task since they can effectively learn meaningful hidden features without human intervention. We follow this approach and propose a new model which effectively solves the task.

We briefly introduce our prior work (Tran et al., 2019) on a fully supervised relation extraction task. Most previous neural network models only exploit one of the following structures to represent relation instances: raw word sequences (Zhou et al., 2016; Wang et al., 2016) and dependency trees (Wen, 2017; Le et al., 2018). While raw sequences can provide all the information of relation

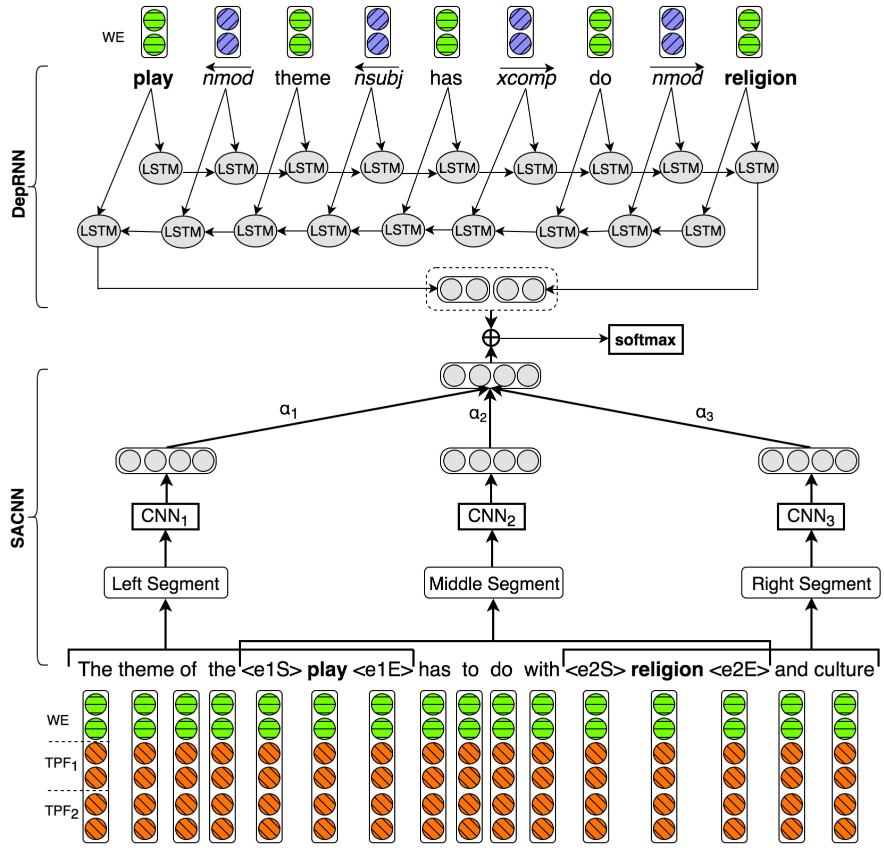

Figure 2.2: Our model for fully supervised relation extraction.

instances, they also add noise to the models from redundant information. While dependency tree structures help the models focus on the concise information captured by the shortest dependency path (SDP) between two entities, they lose some supplementary context in the raw sequence. It is clear that the raw sequence and SDP highly complement each other. We, therefore, combine them to be more effective in determining the relation without losing any information.

The architecture of our model is presented in Figure 2.2. First, we combine Entity Tag Feature (ETF) (Qin et al., 2016) and Tree-based Position Feature (TPF) (Yang et al., 2016) to improve the semantic information between the two entities in the raw input sentences. Then, we propose Segment-Level Attention-based Convolutional Neural Networks (SACNN), which automatically pay special attention to the critical text segments from the raw sentence for relation classifi-

Model	Input	F1
CNN	Original Sentence	83.5
CNN	Middle Segment	84.1
Segment-Level Attention-based CNN	Three Segments	**85.1**

Table 2.1: Effectiveness of the segment-level attention.

Model	F1
Dependence-based RNN	83.8
Segment-Level Attention-based CNN	85.1
Combined	**85.8**

Table 2.2: Evaluation of our combined model.

cation. While the SACNN can learn local features, it cannot handle long-distance dependency between two entities. Meanwhile, the RNN could tackle the problem of long-distance pattern learning (Zhang and Wang, 2015). Besides, the SDP naturally offers the relative positions of subjects and objects through the path directions (Xu et al., 2015). We, therefore, exploit SDP based on the RNN to gain the information in the directional relation. Finally, we combine the SACNN and the DepRNN models to exploit their distinct advantages fully.

We evaluate our model on the benchmark dataset SemEval-2010 Task 8 (Hendrickx et al., 2010). We first examine the segment-level attention mechanism of the SACNN. In Table 2.1, with the same input features, the segment-level attention mechanism makes a great contribution by increasing the $F1$ score by 1 point. Furthermore, to check the effect of combining the SACNN and the DepRNN, in Table 2.2, we compare the performance of each model to our combined model. First, the SACNN's performance is superior to the DepRNN. One possible reason is that while the SACNN selectively focuses on the essential segments and gains local features from the raw sentences, the DepRNN based on the SDP in the SemEval2010 Task 8 dataset can only provide the entity's roles (subject or object) effectively. Then, by combining the SACNN and the DepRNN, our model can exploit the vital information and achieve the best performance.

2.1.3 Zero-Shot Relation Extraction

Although neural network models (Tran et al., 2019; Pouran Ben Veyseh et al., 2020; Tian et al., 2021) for the fully supervised relation extraction task have achieved excellent performance, these models cannot recognize new (unseen) relations that have never been seen in the training process. When putting relation extraction in real-world scenarios where many *new relations* always exist, the current supervised models cannot recognize new relations because they are unobserved during training. Therefore, it is worth inventing models capable of identifying *new relations* that have never been observed before. This task is called zero-shot relation extraction (ZSRE), where a model is trained on labeled instances of the seen relations but then targeted to predict unseen relations for testing instances. Additionally, information on all unseen relations is at the testing stage, including their labels (required information) and descriptions (optional information). Although the ZSRE task is essential for extracting new relations in real-world scenarios, relevant studies on ZSRE are still limited. Thus, we try to improve task performance by proposing effective methods in this study.

We follow the exact definition of ZSRE from previous works (Chen and Li, 2021; Gong and Eldardiry, 2021) to introduce the task. Let $\mathcal{Y}_{\mathcal{S}} = \{y_s^1, \ldots, y_s^n\}$ and $\mathcal{Y}_{\mathcal{U}} = \{y_u^1, \ldots, y_u^m\}$ denote the sets of *seen* and *unseen* relation labels, respectively, where $n = |\mathcal{Y}_{\mathcal{S}}|$ and $m = |\mathcal{Y}_{\mathcal{U}}|$ denote the numbers of relations in the two sets. These two sets are disjoint, *i.e.*, $\mathcal{Y}_{\mathcal{S}} \cap \mathcal{Y}_{\mathcal{U}} = \emptyset$. Given a training set with N samples, the i^{th} sample comprises the input instance X_i, the entities e_{i1} and e_{i2}, and description D_i of the corresponding *seen* relation label $y_s^i \in \mathcal{Y}_{\mathcal{S}}$, hereby denoted as $\{S_i = (X_i, e_{i1}, e_{i2}, D_i, y_s^i)\}_{i=1}^N$. Note that, while relation label information is compulsory, relation description information is optional according to its availability. Using the training set, our goal is to train a model $\mathcal{M}$, *i.e.*, $\mathcal{M}(S_i) \rightarrow y_s^i \in \mathcal{Y}_{\mathcal{S}}$. In the testing stage, given a testing instance S' with two entities, and all *unseen* relation labels in $\mathcal{Y}_{\mathcal{U}}$ (required information) and their descriptions (optional information), $\mathcal{M}$ predicts the *unseen* relation label $y_u^j \in \mathcal{Y}_{\mathcal{U}}$ for S'.

We give an example of the ZSRE task in Table 2.3. In the training stage, the set $\mathcal{Y}_{\mathcal{S}}$ of seen relation labels is {mother, mountain range, member of}. Meanwhile, the unseen relation label set $\mathcal{Y}_{\mathcal{U}}$:{residence, successful candidate} is for the testing

	Input Instance	Relation Label	Relation Description
Training	Jinnah and his wife [Rattanbai Petit]$_{e2}$ had separated soon after their daughter [Dina Wadia]$_{e1}$ was born.	mother	female parent of the subject
	It is approximately 8 km away from [Mount Korbu]$_{e1}$, the tallest mountain of the [Titiwangsa Mountains]$_{e2}$.	mountain range	range or subrange to which the geographical item belongs
	South Africa is part of the [IBSA Dialogue Forum]$_{e2}$, alongside [Brazil]$_{e1}$ and India.	member of	organization or club to which the subject belongs
Testing	In 1959, along with his family, [Gene Chen]$_{e1}$ moved to the USA and settled in [San Francisco]$_{e2}$.	residence	the place where the person is or has been, resident
	In the [1982 General Election]$_{e2}$, [Sir Anerood Jugnauth]$_{e1}$ (SAJ) coalition was elected, he became Prime Minister.	successful candidate	person(s) elected after the election

Table 2.3: Example of the ZSRE task with the training and testing stages. Each input instance contains two entities (*e1* and *e2*) and expresses their semantic relation. The seen relation set $\mathcal{Y}_{\mathcal{S}}$:{mother, mountain range, member of} is for the training stage and the unseen relation set $\mathcal{Y}_{\mathcal{U}}$:{residence, successful candidate} is for the testing stage.

stage. The two sets are disjoint, *i.e.*, $\mathcal{Y}_{\mathcal{S}} \cap \mathcal{Y}_{\mathcal{U}} = \emptyset$. For simplicity, we provide only one labeled instance for each seen relation type in the set $\mathcal{Y}_{\mathcal{S}}$ in the training phase, although it may be many training labeled instances provided for each seen relation type in fact. Additionally, the descriptions of all seen and unseen relations are available from open-source Wikidata[2]. Using the training data, which includes labeled training instances and the information on all seen relations, we train a model $\mathcal{M}$. In the testing phase, the model $\mathcal{M}$ will predict the unseen relation type for each given testing instance. For example, given the testing instance: "*In 1959, along with his family, [Gene Chen]$_{e1}$ moved to the USA and settled in [San Francisco]$_{e2}$.*", $\mathcal{M}$ is expected to predict unseen relation: "residence".

2.1.4 End-to-end Relation Extraction

Another supervised approach for relation extraction task that we focus on is end-to-end relation extraction. Given an unstructured text, it aims to extract pairs of entities with semantic relations to create relational triplets, in the form of (*head*, relation, *tail*). For example, given the unstructured text: "John Smiths lives and works in Paris, the capital and an administrative division of France.",

[2] https://www.wikidata.org/wiki/Wikidata:Main_Page

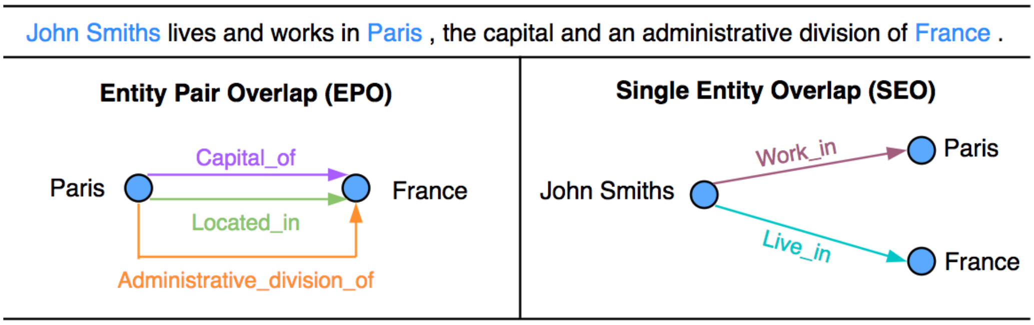

Figure 2.3: An example of the *entity pair overlap* (EPO) and *single entity overlap* (SEO) triplets.

it is expected to extract all relational triplets: {("*Paris*", "Capital_of", "*France*"), ("*Paris*", "Administrative_division_of", "*France*"), ("*Paris*", "Located_in", "*France*"), ("*John Smiths*", "Work_in", "*Paris*"), ("*John Smiths*", "Live_in", "*France*")}. The relational triplets extraction has attracted considerable research effort as it plays a vital role in many NLP applications such as knowledge graph construction (Tran et al., 2021) and question answering (Hao et al., 2017).

One of the biggest challenges of this task is the *overlapping triplet problem*, which is expressed in two scenarios: *entity pair overlap* (**EPO**) and *single entity overlap* (**SEO**). Specifically, EPO occurs when triplets share the same entity pair but with different relations, such as: ("*Paris*", "Capital_of", "*France*"), ("*Paris*", "Located_in", "*France*"), and ("*Paris*", "Administrative_division_of", "*France*"), as shown in Figure 2.3. SEO occurs when two relational triplets share only one common entity, such as: ("*John Smiths*", "Work_in", "*Paris*") and ("*John Smiths*", "Live_in", "*France*").

Most previous works could not efficiently address the *overlapping triplet problem*. This encourages us to consider this problem and propose a new method to solve it productively. The detail of our proposed method for the end-to-end relation extraction task is presented in Chapter 5.

2.2 Related Work

In this section, we introduce related work on *fully supervised relation extraction*, *zero-shot relation extraction*, and *end-to-end relation extraction* in turn.

2.2.1 Related Work on Fully Supervised Relation Extraction

As introduced, this task is naturally treated as a supervised classification problem. Traditional approaches for this task usually rely on hand-crafted features or elaborately designed kernels.

Feature-based methods were firstly used for the relation extraction task among classical supervised machine learning approaches. They rely on lexical, syntactic, and semantic information of an entity pair and their corresponding context. The features include entity mentions, context words, base phrase chunking, part-of-speech (POS) tags, syntactic parse tree, and dependency tree (Kambhatla, 2004b; Zhou et al., 2005). Besides, the tree-based features were also exploited to solve the RE task in some works (Nguyen et al., 2007; Jiang and Zhai, 2007). Other linguistic features were also utilized for the RE. For example, word clusters were used to group similar words into the same cluster (Chan and Roth, 2010; Sun et al., 2011). In addition, several attempts were proposed to exploit entity information such as semantic entity categories (Zhou et al., 2005; Roth and Yih, 2007) and entity statistics from the Web and Wikipedia (Rosenfeld and Feldman, 2007; Chan and Roth, 2010).

In another approach, kernel-based methods aim to design kernels elaborately, which help explore the original representation of a given sentence. They compute similarities between representations by kernel functions performing on subsequences, entire sequences, and grammatical structures such as constituent trees and dependency trees. A popular kernel-based method for RE is sequence kernels that evaluate similar subsequences between sentences. Inspired by Lodhi et al. (2002), Mooney and Bunescu (2005) utilized different types of subsequence patterns such as words before, between and after relation arguments for the RE. Besides, tree-based kernels were also proposed for solving the RE. Zelenko et al. (2003) introduced a tree-based kernel performing on base phrase chunking information. Bunescu and Mooney (2005) presented a new kernel for RE based on the shortest dependency path (SDP) between the two relation entities in the dependency graph. They demonstrated that using SDP yielded significantly higher performance than previous subtree approaches. Zhang et al. (2006a) used a convolutional tree kernel to explore multiple tree representations constructed from the constituent tree structure of a sentence for RE. Several studies made efforts to solve RE by adding richer features into the tree or modifying kernel functions (Khayyamian et al., 2009; Sun et al., 2014).

Recently, neural network models have dominated the work on fully supervised relation extraction task since they can effectively learn meaningful hidden features without human intervention. Zeng et al. (2014) proposed position features to capture target entity information in the sentence. These position features are the relative distances of each word to two entities, which are mapped into continuous-valued vectors, also called position embeddings. Zeng et al. (2015) developed a method of piecewise max pooling and incorporate multi-instance learning into convolutional neural networks for distant supervised relation extraction. Zeng et al. (2017) built inference chains between two target entities via intermediate entities, and proposed a path-based neural relation extraction model to encode the relational semantics from both direct sentences and inference chains. Zhang et al. (2017) presented an entity position-aware attention mechanism in an long short-term memory model to focus on important context words. Guo et al. (2019) proposed attention mechanisms to softly prune the dependency for solving the RE.

More recently, with the appearance of pretrained language models, performance on a wide range of NLP downstream tasks have been significantly improved, including relation extraction. Baldini Soares et al. (2019a) simply inserted entity marker tokens in the original sentence to indicate entity positions and inputted it to a BERT-based model for classifying the relation type. Zhang et al. (2019) enhanced language representation with external knowledge by incorporating informative entities in knowledge graphs, thereby improving the performance on the related downstream tasks such as named entity recognition and relation extraction. Wang et al. (2019) built upon BERT with an entity-aware self-attention mechanism to integrate information from all entity pairs in a sentence. Zhou and

Chen (2021) improved the baseline methods for RE by revisiting two problems that affect the performance of existing relation classifiers, namely entity representation and noisy or ill-defined labels.

2.2.2 Related Work on Zero-Shot Relation Extraction

Only a few relevant studies have been conducted on zero-shot relation extraction (ZSRE). Levy et al. (2017) regarded ZSRE as a question-answering task. They first manually defined 10 question templates to represent each relation type and then made predictions by training a reading comprehension model to determine which relation satisfies the given instance and question. Because this method requires human effort to define question templates for unseen relations, it is possibly unfeasible and impractical to prepare such templates for multiple new unseen relations in real-world scenarios. Obamuyide and Vlachos (2018) formulated ZSRE as a textual entailment task, where the input instance with two entities is the premise P, whereas the relation description is the hypothesis H. They then used existing textual entailment models and required the models to predict whether P matches H. Specifically, they adopted the enhanced sequential inference model (ESIM) (Chen et al., 2017) and conditioned inference model (CIM) (Rocktäschel et al., 2016) as their base models.

Recently, Chen and Li (2021) presented a model called ZS-BERT, which learns two functions to project sentences and relation descriptions into an embedding space by jointly minimizing the distances between them and classifying the seen relations. ZS-BERT then uses the nearest neighbor search to obtain the prediction of unseen relations, although this technique is prone to suffering from the hubness problem (Radovanovic et al., 2010). Another severe problem is relation representations generated by feeding their relation descriptions into the frozen pre-trained Sentence-BERT (Reimers and Gurevych, 2019). These relation representations were fixed during the training. This hinders the learning of meaningful relation representations, thereby affecting task performance.

Gong and Eldardiry (2021) proposed a prompt-based model with semantic knowledge augmentation (ZS-SKA) to recognize unseen relations under the zero-shot setting. They generated augmented instances with unseen relations from instances with seen relations following a new word-level sentence translation rule.

By creating representations of the seen and unseen relations with augmented instances and prompts through prototypical networks, the distance between each query instance and all prototype embeddings of all relations are calculated for prediction. This approach requires the provision of unseen relation labels and external knowledge graphs during the training phase. Thus, it is impractical and infeasible in real-world scenarios because the required information is not readily available at the training stage.

More recently, Wang et al. (2022) proposed a novel Relation Contrastive Learning framework (RCL) to mitigate above two types of similar problems: Similar Relations and Similar Entities. By jointly optimizing a contrastive instance loss with a relation classification loss on seen relations, RCL can learn subtle difference between instances and achieve better separation between different relation categories in the representation space simultaneously. Especially in contrastive instance learning, the dropout noise as data augmentation is adopted to amplify the semantic difference between similar instances without breaking relation representation, so as to promote model to learn more effective representations. Experimental results on two benchmark datasets demonstrated the effectiveness of their framework.

2.2.3 Related Work on End-to-end Relation Extraction

Researchers have made great efforts to extract relational triplets from unstructured text, which can be directly used for automatic knowledge graph construction. There are two main methods for solving this task, namely pipeline methods and joint learning methods.

Early works (Choi et al., 2006; Yang and Cardie, 2013; Singh et al., 2013) regarded the joint extraction task in a pipeline manner. They extracted relational triplets in two isolated steps, firstly identifying entities, and then classifying the relations between entities.

Choi et al. (2006) employed linear-chain Conditional Random Fields (CRFs) to develop two separate token-level sequence-tagging classifiers for the entity recognition. The sequence-tagging classifiers were trained using only local syntactic and lexical information to extract each type of entity without knowledge of any nearby or neighboring entities or relations. Besides, they also developed a relation classifier using Markov order-0 CRFs, which is trained using only local syntactic information potentially useful for connecting a pair of entities, but has no knowledge of nearby or neighboring extracted entities and link relations. However, the entity recognition and the relation extraction are separated to train. Thus, these methods cannot transform the internal association between entities and relations into contextual information that should be integrated into methods. Yang and Cardie (2013) presented a joint inference model based on conditional random field (CRF) to get the optimal prediction for both entity recognition and relation extraction. Specifically, the proposed model leveraged knowledge from predictors that optimizes subtasks with constraints of enforcing global consistency to seek the optimal solution. Singh et al. (2013) developed a joint probabilistic graphical model to construct a circular pipeline consisting of entity tagging, relation extraction, and coreference. Since the resulting model has a high tree-width and contains a large number of variables, they also presented a novel extension to belief propagation that sparsifies the domains of variables during inference. More recently, Zhong and Chen (2021) introduced a simple and pipelined approach for entity and relation extraction and established the new state-of-the-art on standard benchmarks. Their approach essentially learns independent two encoders for entity recognition and relation extraction and merely uses the entity model to construct the input for the relation model. They also presented an efficient approximation, obtaining a large speedup at inference time with a small reduction in accuracy. Although these pipeline methods are quite simple, they often suffer from the error propagation problem and ignore the relevance between the two steps.

To ease the two issues above, subsequent works attempted to build joint learning models that learn entities and relations simultaneously in a single manner. They can be divided into two main approaches: feature-based models (Yu and Lam, 2010; Li and Ji, 2014; Miwa and Sasaki, 2014; Ren et al., 2017) and neural network-based models (Gupta et al., 2016; Katiyar and Cardie, 2017; Zheng et al., 2017; Zeng et al., 2018; Fu et al., 2019; Yu et al., 2020). The former rely heavily on feature engineering and require intensive manual efforts, whereas the latter are mainly based on neural network architectures. Zheng et al. (2017) introduced a unified tagging scheme and converted the joint entity and relation extraction task to an end-to-end sequence tagging problem. This method can directly model relational triplets as a whole at the triplet level because the unified tagging scheme already integrates the information of both entities and relations. However, most previous studies ignored the problem of overlapping relational triplets. Zeng et al. (2018) presented three patterns of overlapping triplets and made an effort to address the problem via a sequence-to-sequence model with a copy mechanism. Subsequently, Fu et al. (2019) also focused on this problem and proposed a GCN-based method to address it. Recently, Yu et al. (2020) introduced a novel decomposition strategy that decomposes the task into *HE* and *TER* extractions, where the *HE* extractor detects head entities and the *TER* extractor identifies the corresponding tail entity and relation for each given HE. Although this approach significantly outperforms previous works, it still cannot solve the *entity pair overlap* problem as Yu et al. (2020) stated in their work. Yuan et al. (2020) proposed a relation-attentive sequence labeling framework named RSAN for joint entity and relation extraction. It decomposes the overlapping triplets extraction problem into several relation-specific entity tagging processes, and applies attention mechanism to incorporate finegrained relational information as the guidance of entity extraction.

On a related note, pre-trained language models have also been exploited for entity and relation extraction, thereby utilizing prior knowledge and achieving superior results. Zhao et al. (2020) adopted BERT (Devlin et al., 2019) as the machine reading comprehension model to solve the joint entity and relation extraction. Wang and Lu (2020) introduced table sequence encoders architecture for joint extraction of entities and their relations. It learns two separate encoders rather than one – a sequence encoder and a table encoder where explicit interactions exist between the two encoders. They also presented a new method to effectively employ useful information captured by the pre-trained language models for such a joint learning task where a table representation is involved. Hang et al. (2021) proposed BERT-JEORE, an end-to-end neural network model that is based on BERT for the joint extraction of entities and overlapping relations. They used source-target BERT to generate an entity label for each token in the sample and utilized an overlapping relation extraction model to create an unlimited number of relational triplets. Shang et al. (2022) proposed novel joint entity and relation extraction model, named OneRel, which casts joint extraction as a fine-grained triple classification problem. Specifically, their model consists of a scoring-based classifier and a relation-specific horns tagging strategy. The former evaluates whether a token pair and a relation belong to a factual triple. The latter ensures a simple but effective decoding process.

Chapter 3

Improving Discriminative Learning for Zero-Shot Relation Extraction

3.1 Introduction

As introduced in Section 2.1.3, the zero-shot relation extraction (ZSRE) task is essential for extracting new relation in real-world scenarios. However, relevant studies on ZSRE are still limited. Levy et al. (2017) tackled this task by using reading comprehension models with predefined question templates. Obamuyide and Vlachos (2018) simply reduced ZSRE to a text entailment task, utilizing existing textual entailment models. Recently, Chen and Li (2021) presented ZS-BERT, which projects sentences and relations into a shared space and uses the nearest neighbor search to predict unseen relations.

The previous studies overlooked the importance of learning discriminative embeddings. In essence, discriminative learning helps models distinguish relations better, especially on similar ones. Our study focuses on this aspect and demonstrates its significance for improving ZSRE. Specifically, we propose a new model incorporating discriminative embedding learning (Han et al., 2021) for both sentences and semantic relations, which is inspired by contrastive learning (Chen et al., 2020) commonly used in computer vision. In addition, instead of using distance metrics to predict unseen relations as done by Chen and Li (2021), we use a self-adaptive comparator network to judge whether the relationship between a sentence and a relation is consistent. This verification process helps the model to learn more discriminative embeddings. Experimental results on two datasets showed that our method significantly outperforms the existing methods for ZSRE.

3.2 Proposed Model

3.2.1 Framework

Sentence Encoder. We use BERT (Devlin et al., 2019) as the basic encoder to generate contextualized representations of input sentences. Following Baldini Soares et al. (2019b), we first augment each input sentence with four reserved word pieces ([E1], [/E1], [E2], and [/E2]) to indicate two entities in the input sentence. For example, in the upper part of Figure 3.1, the input sentence is "*[Amazon]$_{e1}$ was founded by [Jeff Bezos]$_{e2}$ in 1994.*" becomes "*[E1] Amazon [/E1] was founded by [E2] Jeff Bezos [/E2] in 1994.*". Then, we tokenize the input sentence with word-piece tokenization (Sennrich et al., 2016). Two special tokens [CLS] and [SEP] are appended to the first and last positions, respectively. After that, we input them through a pre-trained BERT encoder (Devlin et al., 2019). Finally, we obtain the vector representing the semantic relationship between the two entities by concatenating the two hidden state vectors of the two start tokens ([E1] and [E2]).

Relation Encoder. Most relations are well defined, and their descriptions are available from open resources such as Wikidata[1] (Chen and Li, 2021). However, if relation descriptions are not available in a new domain, we can easily create the necessary relation descriptions manually by humans, as it does not require much effort. Therefore, for each relation, we feed its corresponding relation description into a pre-trained Sentence-BERT encoder (Reimers and Gurevych, 2019) and obtain the representation vector using the mean pooling operation on the outputs. This procedure is shown in the bottom part of Figure 3.1. The ground truth relation of the example is "*founded by*", along with its description[2] "*Founder or*

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

[2] https://www.wikidata.org/wiki/Property:P112

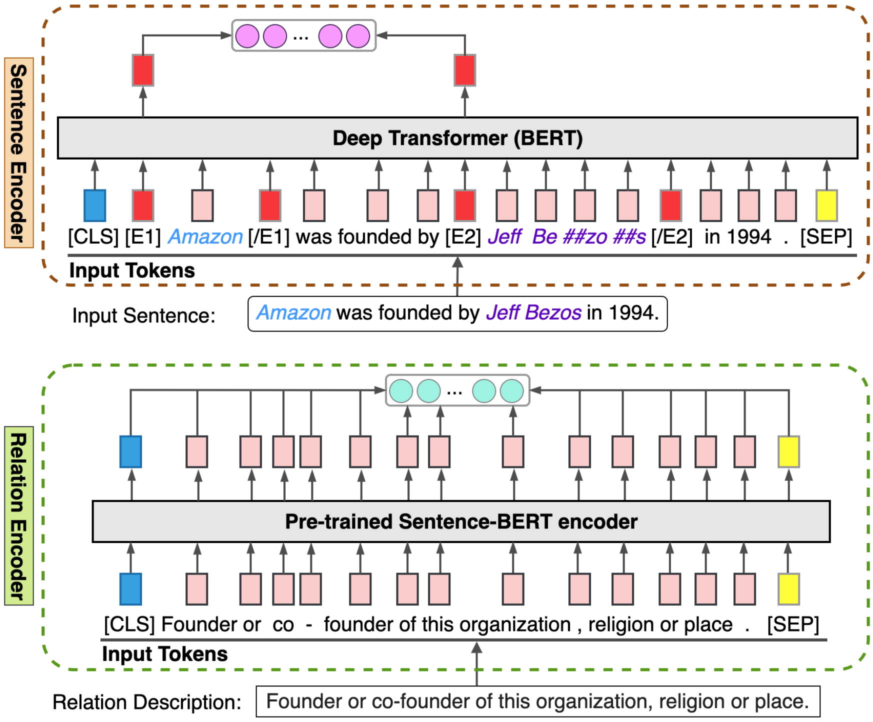

Figure 3.1: **Sentence Encoder** and **Relation Encoder**.

co-founder of this organization, religion or place". This relation description is fed into the Sentence-BERT to obtain the relation representation vector.

Overview of the Model. On the basis of the two modules above, we present our full model in Figure 3.2. Given a training mini-batch of N sentences, we feed them into the **Sentence Encoder** and a subsequent nonlinear projector to obtain N final sentence embeddings. Simultaneously, we acquire K different relations from the N sentences. The K corresponding descriptions of the K relations are then fed into the **Relation Encoder** and a subsequent nonlinear projector to acquire the final relation embeddings. To obtain more discriminative embeddings, we introduce the learning constraints described in detail later. Finally, we concatenate pairs from the two spaces and use a network $\boldsymbol{F}$ to judge whether the relationship between a sentence and a relation is consistent.

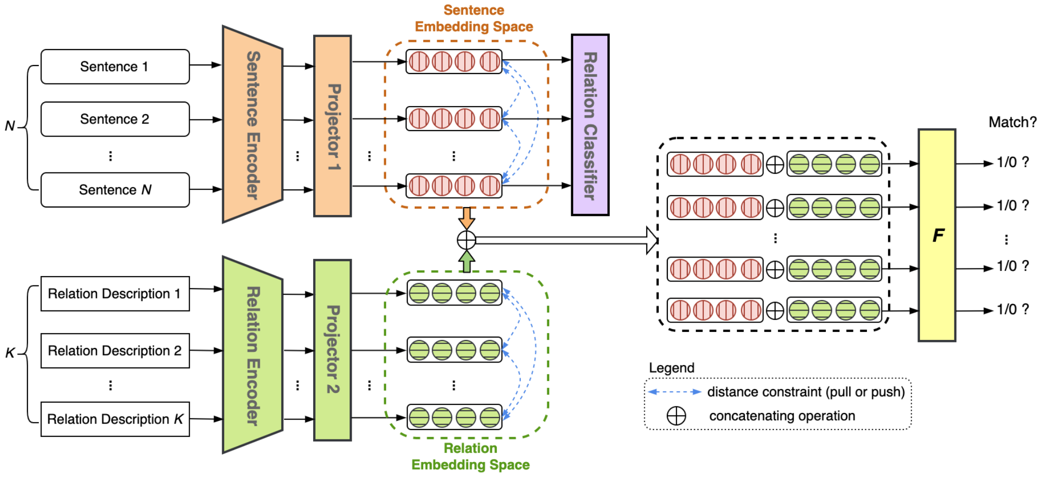

Figure 3.2: Overview of our proposed model with an input training mini-batch of size N.

3.2.2 Model Training

To boost the learning of discriminative embeddings for sentences and relations, we consider three main goals in training: (1) discriminative sentence embeddings, (2) discriminative relation embeddings, and (3) an effective comparator network $\boldsymbol{F}$.

Discriminative Sentence Embeddings. In Figure 3.2, given a mini-batch of N sentences, we obtain N corresponding sentence embeddings: $[\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N]$. To learn the discriminative features, we first use a softmax multi-class relation classifier to predict the seen relation for each sentence:

$$\mathcal{L}_{\text{Softmax}} = -\frac{1}{N}\sum_{i}^{N} y_s^i \log(\hat{y_s}^i), \tag{3.1}$$

where $y_s^i \in \mathcal{Y}_{\mathcal{S}}$ is the ground-truth seen relation label of the i^{th} sentence and $\hat{y_s}^i$ is the predicted probability of y_s^i. However, such a softmax loss only encourages the separability of the inter-class features. Meanwhile, discriminative power characterizes features in both the separable inter-class differences and the compact intra-class variations (Wen et al., 2016). Thus, we use another loss to ensure the intra-class compactness. First, the similarity distance between two sentences is given by

$$d(\mathbf{s}_i, \mathbf{s}_j) = 1/(1 + \exp(\frac{\mathbf{s}_i}{\|\mathbf{s}_i\|} \cdot \frac{\mathbf{s}_j}{\|\mathbf{s}_j\|})). \tag{3.2}$$

Clearly, this value should be small for any sentence pair of the same relation (*positive* pair) and large for a *negative* pair. We then apply such distance constraints on all T unordered sentence pairs, where $T = N(N-1)/2$, and formulate the loss as

$$\mathcal{L}_{\mathrm{S2S}} = -\frac{1}{T} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \Big(\mathbb{I}_{ij} \log d(\mathbf{s}_i, \mathbf{s}_j) + (1 - \mathbb{I}_{ij}) \log(1 - d(\mathbf{s}_i, \mathbf{s}_j)) \Big), \tag{3.3}$$

where $\mathbb{I}_{ij} = 1$ if the pair $(\mathbf{s}_i, \mathbf{s}_j)$ is *positive* or 0 otherwise. $\mathcal{L}_{\mathrm{S2S}}$ not only ensures the intra-relation compactness but also encourages the inter-relation separability further. Finally, the final loss of learning discriminative sentence embeddings in the sentence embedding space is defined as follows:

$$\mathcal{L}_{sent} = \mathcal{L}_{Softmax} + \gamma \cdot \mathcal{L}_{\mathrm{S2S}}, \tag{3.4}$$

where γ is a hyperparameter. With this joint supervision, it is expected that not only the inter-class sentence embedding differences are enlarged, but also the intra-class sentence embedding variations are reduced. Thus, the discriminative power of the learned sentence embeddings will be enhanced.

Discriminative Relation Embeddings. In Figure 3.2, we obtain K corresponding relation embeddings: $[\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_K]$ for K different relations in the relation embedding space. From the K relations, we have a total of Q pairs ($Q = K(K-1)/2$), where each pair includes two different unordered relations. Thus, we maximize distance for each of these pairs and define the loss of learning discriminative relation embeddings by

$$\mathcal{L}_{rel} = -\frac{1}{Q} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \log(1 - d(\mathbf{r}_i, \mathbf{r}_j)), \tag{3.5}$$

where $d(\mathbf{r}_i, \mathbf{r}_j)$ is the similarity distance between two relations using Equation 3.2.

Comparator Network. After obtaining the discriminative embeddings of sentences and relations, we use a comparator network $\boldsymbol{F}$ to judge how well a sentence is consistent with a specific relation. This validation information will guide our model to learn more discriminative embeddings. In Figure 3.2, we concatenate sentences and relations as pairs and feed them into $\boldsymbol{F}$. To enhance the training efficiency, we control the rate of positive and negative pairs. Specifically, we keep all positive pairs but randomly select only a part of negative pairs (e.g., positive:negative *rate* is 1:3). $\boldsymbol{F}$ is a two-layer nonlinear neural network that outputs a scalar similarity score in the range of (0,1]. Finally, the loss of training $\boldsymbol{F}$ is defined as

$$\mathcal{L}_F = -\frac{\sum_{i=1}^{\mathcal{N}_{pos}} \log v_i + \sum_{j=1}^{\mathcal{N}_{neg}} \log\left(1 - v_j\right)}{\mathcal{N}_{pos} + \mathcal{N}_{neg}}, \tag{3.6}$$

where v_i and v_j are the similarity scores of the i^{th} positive pair and j^{th} negative pair, respectively; $\mathcal{N}_{pos}$ and $\mathcal{N}_{neg}$ are the number of positive pairs and negative pairs for training.

Total Loss. Based on the three aforementioned losses, the full loss function for training our model is as follows:

$$\mathcal{L} = \mathcal{L}_F + \alpha\mathcal{L}_{sent} + \beta\mathcal{L}_{rel}, \tag{3.7}$$

where α and β are hyperparameters that control the importance of $\mathcal{L}_{sent}$ and $\mathcal{L}_{rel}$, respectively.

3.2.3 Zero-Shot Relation Prediction

In the testing stage, we conduct zero-shot relation prediction by comparing the similarity score of a given sentence with all the unseen semantic relation representations. We classify the sentence $\mathbf{s}_i$ to the unseen relation that has the largest similarity score among relations, which can be formulated as

$$P_{zsre}\left(\mathbf{s}_i\right) = \max_j \left\{v_{ij}\right\}_{j=1}^{|\mathcal{Y}_\mathcal{U}|}. \tag{3.8}$$

3.3 Experiments

3.3.1 Dataset

Following the previous work (Chen and Li, 2021), we evaluate our model on two benchmark datasets: **Wiki-ZSL** and **FewRel** (Han et al., 2018). FewRel is a human-annotated balanced dataset consisting of 80 relation types, each of which has 700 instances. Wiki-ZSL is a subset of Wiki-KB dataset (Sorokin and Gurevych, 2017), which filters out both the "none" relation and relations that appear fewer than 300 times. The statistics of Wiki-KB, Wiki-ZSL, and FewRel are shown in Table 3.1. Note that descriptions of the relations in the above datasets are available and accessible from the open source Wikidata[3].

	#instances	#relations	avg. len.
Wiki-KB	$1,518,444$	354	23.82
Wiki-ZSL	$94,383$	113	24.85
FewRel	$56,000$	80	24.95

Table 3.1: The statistics of the datasets.

3.3.2 Experimental Settings

Following Chen and Li (2021), we randomly selected m relations as unseen ones ($m = |\mathcal{Y}_{\mathcal{U}}|$) for the testing set and split the entire dataset into the training and testing datasets accordingly. This guarantees that the m relations in the testing dataset do not appear in the training dataset. We used macro precision (P), macro recall (R), and macro F1-score ($F1$) as the evaluation metrics.

We implemented the neural networks using the PyTorch library[4]. The *tanh* function is used with each nonlinear projector in our model. The comparator network $\boldsymbol{F}$ is a two-layer nonlinear neural network in which the hidden layer is equipped with the *tanh* function, and the output layer size is outfitted with the *sigmoid* function. The dropout technique was applied at a rate of 0.3 on the

[3] https://www.wikidata.org/wiki/Wikidata:Main_Page

[4] PyTorch is an open-source software library for machine intelligence: https://pytorch.org/

hidden layer and embeddings of sentences and relations in the two embedding spaces. We used Adam (Kingma and Ba, 2015) as the optimizer, in which the initial learning rate was $5e-6$; the batch size was 16 on FewRel and 32 on Wiki-ZSL; and $\alpha = 0.7$, $\beta = 0.3$, and $\gamma = 0.5$.

3.3.3 Results and Analysis

Main Result. The experimental results obtained by varying m unseen relations are shown in Table 3.2. It can be observed that our model steadily outperforms the competing methods on the test datasets when considering different values of m. In addition, the improvement in our model is smaller when m is larger. There are two possible reasons for this phenomenon. First, following the experiment settings of the ZSRE, since the whole dataset of N relations is divided into train set ($(N-m)$ seen relations) and testing set (m unseen relations), the number of seen relations for the training phase will be smaller when m is larger. Second, an increase in m also leads to a rise in the possible choices for prediction, thereby making it more difficult to predict the correct unseen relation. We plan to overcome this disadvantage in our future work. We will propose new models that improve the model robustness to solve ZSRE effectively in case of the limited training dataset.

	Wiki-ZSL			FewRel		
$m = 5$	*P*	*R*	*F1*	*P*	*R*	*F1*
ESIM*	48.58	47.74	48.16	56.27	58.44	57.33
CIM*	49.63	48.81	49.22	58.05	61.92	59.92
ZS-BERT*	71.54	72.39	71.96	76.96	78.86	77.90
ZS-BERT†	74.32	71.72	72.97	80.96	78.00	79.44
Ours	**87.48**	**77.50**	**82.19**	**87.11**	**86.29**	**86.69**
$m = 10$	*P*	*R*	*F1*	*P*	*R*	*F1*
ESIM*	44.12	45.46	44.78	42.89	44.17	43.52
CIM*	46.54	47.90	45.57	47.39	49.11	48.23
ZS-BERT*	60.51	60.98	60.74	56.92	57.59	57.25
ZS-BERT†	64.53	58.30	61.23	60.13	55.63	57.80
Ours	**71.59**	**64.69**	**67.94**	**64.41**	**62.61**	**63.50**
$m = 15$	*P*	*R*	*F1*	*P*	*R*	*F1*
ESIM*	27.31	29.62	28.42	29.15	31.59	30.32
CIM*	29.17	30.58	29.86	31.83	33.06	32.43
ZS-BERT*	34.12	34.38	34.25	35.54	38.19	36.82
ZS-BERT†	35.42	33.47	34.42	39.09	36.70	37.84
Ours	**38.37**	**36.05**	**37.17**	**43.96**	**39.11**	**41.36**

Table 3.2: Results with different m values in percentage. * indicates the results reported by Chen and Li (2021); † marks the results we reproduced using the official source code of Chen and Li (2021).

$m = 5$	*F1*	
	Wiki-ZSL	FewRel
Ours	**82.19**	**86.69**
Ours w/o $\mathcal{L}_{sent}$ ($\alpha = 0$)	74.42	81.05
Ours w/o $\mathcal{L}_{rel}$ ($\beta = 0$)	78.92	84.27
Ours w/o $\mathcal{L}_{S2S}$ ($\gamma = 0$)	77.13	82.95

Table 3.3: Ablation study.

Obamuyide and Vlachos (2018) simply used two basic text entailment models (ESIM and CIM) that may not be entirely relevant for ZSRE. Besides, they ignored the importance of discriminative feature learning for sentences and relations. Chen and Li (2021) also overlooked the necessity of learning discriminative embeddings. In addition, the nearest neighbor search method in ZS-BERT is prone to cause the hubness problem (Radovanovic et al., 2010). Thus, our model was designed to overcome the existing limitations. Compared with ZS-BERT, our model significantly improved its performance when $m = 5$, by 9.22 and 7.25 $F1$-score on Wiki-ZSL and FewRel, respectively.

Impact of Discriminative Learning. To gain more insight into the improvement in our model, we analyzed the importance of learning discriminative features in both the sentence and relation spaces. In Table 3.3, we consider three special cases of Equation 3.7: (1) $\alpha = 0$ means no $\mathcal{L}_{sent}$; (2) $\beta = 0$ means no $\mathcal{L}_{rel}$; and (3) $\gamma = 0$ means no $\mathcal{L}_{S2S}$, which is a part of $\mathcal{L}_{sent}$. Clearly, all three losses are important for training our model to obtain the best performance. In particular, $\mathcal{L}_{sent}$ for learning discriminative sentence features is more important than $\mathcal{L}_{rel}$ for learning discriminative relation embeddings, as the performance decreases significantly after removing it. In addition, $\mathcal{L}_{S2S}$ plays a vital role in $\mathcal{L}_{sent}$ since it mainly ensures the intra-relation compactness property of discriminative sentence embeddings.

Feature Space Visualization. To gain more insights into the quality of sentence embeddings, we visualized the testing sentence embeddings produced by

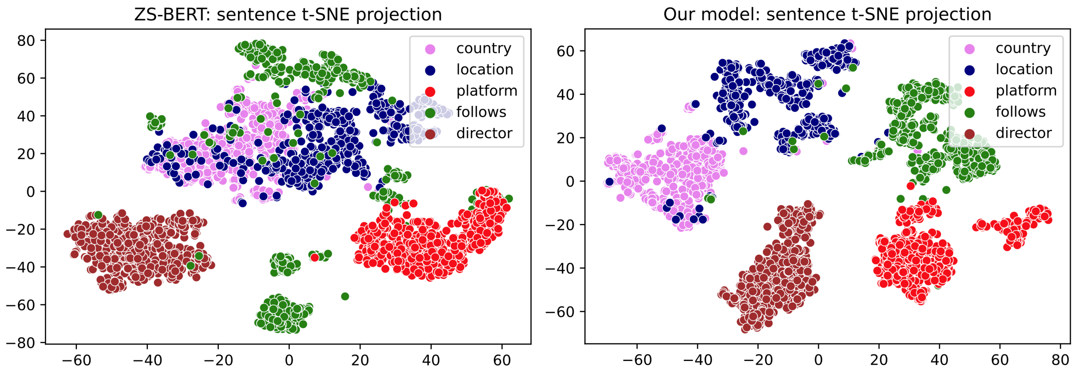

Figure 3.3: Visualization of the sentence embeddings by ZS-BERT and our model when $m = 5$ on the FewRel.

ZS-BERT and our model in the case of $m = 5$ on the FewRel[5] dataset using t-SNE (Maaten and Hinton, 2008). Figure 3.3 shows that the embeddings generated by our model express not only a larger inter-relation separability but also a better intra-relation compactness, compared with the embeddings by ZS-BERT.

Let us focus on two relations: *country*[6] and *location*[7]. According to the descriptions of these two relations, we can see that they are somewhat similar but different in some details. Specifically, an ordered entity pair ($\boldsymbol{e1}$, $\boldsymbol{e2}$) in a sentence expresses the relation "country" if and only if $\boldsymbol{e2}$ must be a country and $\boldsymbol{e2}$ has sovereignty over $\boldsymbol{e1}$. Meanwhile, if the entity pair ($\boldsymbol{e1}$, $\boldsymbol{e2}$) does not hold the relation "country", it may appear the relation "location" when $\boldsymbol{e2}$ is a place that $\boldsymbol{e1}$ happens or exists. Thus, the two similar relations make it difficult for ZS-BERT to distinguish them. Meanwhile, our model can discriminate between them. These observations again prove the necessity of learning discriminative features for ZSRE.

3.4 Conclusion

This chapter presents a new model to solve the ZSRE task. Our model aims to enhance the discriminative embedding learning for both sentences and relations.

[5]The FewRel dataset is annotated by crowdworkers, thereby ensuring high quality.
[6]https://www.wikidata.org/wiki/Property:P17
[7]https://www.wikidata.org/wiki/Property:P276

It boosts inter-relation separability and intra-relation compactness of sentence embeddings and maximizes distances between different relation embeddings. In addition, a comparator network is used to validate the consistency between a sentence and a relation. Experimental results on two benchmark datasets demonstrated the superiority of the proposed model for ZSRE.

Chapter 4

Enhancing Semantic Correlation between Instances and Relations for Zero-Shot Relation Extraction

4.1 Introduction

Zero-shot relation extraction aims to recognize (new) unseen relations that cannot be observed during training. Due to this point, recognizing unseen relations with no corresponding labeled training instances is a challenging task. Meanwhile, information on all unseen relations is given at the testing stage, including their labels (required information) and descriptions (optional information). Thus, to make a correct prediction, the model must profoundly understand the semantic relationship between each instance and all unseen relations. Following this intuitive reasoning, we argue that enhancing the semantic correlation between instances and relations is the key to solving ZSRE effectively.

While relevant studies on ZSRE are still limited, these studies underestimated the key solution above and had some other limitations. For example, Levy et al. (2017) formulated ZSRE as a question-answering task by creating manually predefined question templates for each relation. However, it is infeasible and impractical to make such human efforts for many new unseen relations in the zero-shot setting. Obamuyide and Vlachos (2018) reduced ZSRE to a text entailment task and designed a binary classifier to indicate whether a given relation description depicts the relationship between two entities in an input instance. This approach requires the inefficient execution of multiple binary classifications over all relation descriptions and cannot make relations comparable.

More recently, Chen and Li (2021) presented a model called ZS-BERT, which first projects instances and relations in a shared space and then minimizes the distance between each instance and the corresponding relation. However, although ZS-BERT considers the semantic relationship between instances and relations, it has a severe limitation. Specifically, as relation representations are fixed during training, they lead to low-quality relation representations and hinder grasping the semantic relationship. Gong and Eldardiry (2021) then presented a prompt-based model with semantic knowledge augmentation to recognize unseen relations. They initially generated augmented instances with unseen relations from training instances with seen relations. Then, they designed prompts based on an external knowledge graph to learn representations for both seen and unseen relations. However, this model requires unseen relation labels and an external knowledge graph in the training stage, although such information is not readily available in real-world scenarios. Besides, in Chapter 3, we introduce a new method that improves discriminative learning for ZSRE. However, although this method helps significantly boost task performance, it still has two limitations. First, this method only exploited relation description to create relation representation via a fixed pre-trained Sentence-BERT model. Meanwhile, we can obtain a better relation representation by using both relation label and description via a learnable BERT-based model, thereby expecting to improve the system performance. Second, our prior method only uses a simple comparator network $\boldsymbol{F}$, a two-layer nonlinear neural network, to learn the semantic consistency between sentences and relations. Nevertheless, such a comparator network may not be good enough to grasp deeply the semantic correlation between sentences and relations, which is the key to solving ZSRE effectively.

This chapter proposes a new approach to overcome the limitations of previous studies and our prior method (introduced in Chapter 3). Without any external knowledge graphs or unseen relation labels in the training phase, our model focuses on effectively grasping the semantic correlation between instances and relations because it is a crucial solution for solving the ZSRE. Our model achieves this by concentrating on the following three aspects.

First, our model acquires meaningful and high-quality representations for instances and relations. This aspect plays an essential role in understanding the semantic correlation between instances and relations. Specifically, instead of using fixed pre-trained relation representations, as in the previous work (Chen and Li, 2021), our approach obtains the instance and relation representations via a learnable BERT-based encoder module. We also exploit relation labels and relation descriptions to attain better relation representations.

Second, the previous studies (Chen and Li, 2021; Gong and Eldardiry, 2021) prepared mini-batches in a standard manner, where each training mini-batch comprises some of the labeled instances by a random sampling technique. In contrast to this approach, we design each mini-batch as a mini-task, including K different seen relations and K corresponding instances (K is a hyperparameter), and force the model to pair them exactly. This strategy encourages the model to grasp the semantic relationship between instances and relations deeply.

Finally, the previous studies (Chen and Li, 2021; Gong and Eldardiry, 2021) treat relation representations as targets and minimize the probability distribution from each instance to its corresponding relation in the shared space. This approach is a one-way interaction that cannot fully exploit the semantic relationship between instances and relations. Instead, we use two-way interaction, which grasps the interaction not only "from each instance to relations" but also "from each relation to instances" and constrains the consistency of the two interaction distributions.

The contributions of this chapter are summarized as follows:

(a) We indicate that enhancing the semantic correlation between instances and relations is the key to drastically improving the performance of ZSRE.

(b) We propose an approach that focuses on this goal by learning high-quality relation representation, designing strategic mini-batches, and binding two-way semantic consistency.

(c) Extensive experiments on two benchmark datasets demonstrated the effectiveness and robustness of our approach, as it significantly outperformed the existing state-of-the-art methods.

It can be seen that our proposed model is closely related to dense retrieval models. Specifically, dense retrieval models aim to retrieve relevant documents for a given query in information retrieval research applications. They try to capture the deep semantic relationship between queries and documents in embedding space by mapping documents and queries to k-dimensional real-valued vectors. Existing dense retrieval models can be classified into two categories. One line of research is negative sampling (Karpukhin et al., 2020; Xiong et al., 2021; Zhan et al., 2021), while the other line is knowledge distillation (Qu et al., 2020; Lin et al., 2020; Hofstätter et al., 2021), which adopts a cross-encoder to generate pseudo labels. The negative sampling approach selects several negative documents from the entire corpus for a given training query. Then, the dense retrieval model encodes the queries and documents into embeddings and uses the inner product to compute their relevance scores. The training method uses the scores to compute a pairwise loss based on the gold annotations. Our proposed model for the ZSRE task is close to this method but still different. Concretely, we design each purposeful mini-batch including K relations and K corresponding instance and force the model to pair them exactly. Besides, we also propose to put the added constraint using the KL-Divergence Loss to improve the semantic relationship between instances and relations, which has never been suggested before.

4.2 Approach

This section presents the details of the proposed approach for solving the ZSRE. Figure 4.1 shows the overall learning framework. This model aims to enhance the semantic correlation between instances and relations.

4.2.1 Instance Representation

We use BERT (Devlin et al., 2019) as the basic encoder to generate contextualized representations of input instances. Following Baldini Soares et al. (2019b), we first augment each input instance with four reserved word pieces ([E1], [/E1], [E2], and [/E2]) to indicate two entities in the input instance. For example, the input instance is "*In 1959, along with his family, [Gene Chen]*$_{e1}$ *moved to the USA and settled in [San Francisco]*$_{e2}$*.*" becomes "*In 1959, along with his family, [E1] Gene Chen [/E1] moved to the USA and settled in [E2] San Francisco [/E2] .*".

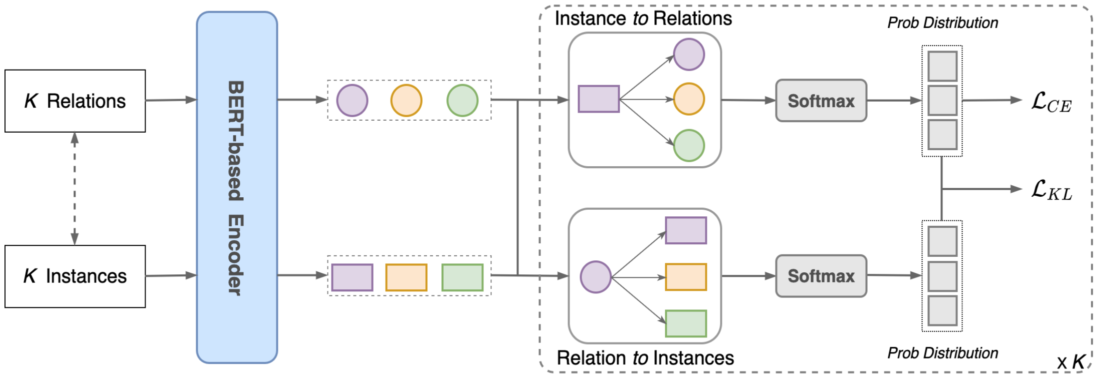

Figure 4.1: The overall framework of our approach. The input is a training mini-batch that consists of K different relations (e.g., $K = 3$) and K corresponding instances. The circles and rectangles are relations and instances, respectively. Distinct colors represent different classes. For simplicity, we illustrate interactions from only one instance to relations and only one relation to instances on the right side. As instance-to-relations classification is the original ZSRE task, then $\mathcal{L}_{CE}$ is used to supervise it. Further, we use $\mathcal{L}_{KL}$ to put an added constraint on the correlation between the two distributions.

We then feed this sequence into the BERT encoder. Finally, the input instance representation is obtained by concatenating two hidden state vectors of the two start tokens ([E1] and [E2]) with the final dimension of $\mathbb{R}^{2d}$.

4.2.2 Relation Representation

A BERT-based encoder is also used to obtain semantic relation representations. As introduced in the ZSRE task definition, while relation label information is compulsory, relation description is optional according to its availability. Thus, if only the relation label is provided, we input it into the BERT encoder. Conversely, when both relation labels and descriptions are given, we initially concatenate them using a special token: [SEP]. For example, the relation label "*residence*" with its description: "*the place where the person is or has been, resident*" becomes "*residence [SEP] the place where the person is or has been, resident*". Then, we feed it into the BERT encoder.

In both cases, the final relation representation is attained by the hidden states corresponding to the [CLS] token (converted to $\mathbb{R}^{2d}$ dimension with a linear transformation). The linear transformation guarantees equal dimension sizes of the instance and relation representations.

4.2.3 Semantic Correlation Learning

Given a training set with N samples of T different *seen* relations ($T < N$), we create mini-batches to train our model. To facilitate the model in grasping the semantic correlation between instances and relations, we intentionally design mini-batches differently. Each mini-batch consists of K different *seen* relations ($K \leq T$), randomly sampled, and K corresponding instances. The model is then required to match instances to the corresponding relations exactly.

We feed each mini-batch into the BERT-based encoder and obtain K relation representations: $\left\{\mathbf{r}_i \in \mathbb{R}^{2d}; i = 1, \ldots, K\right\}$ and K instance representations: $\left\{\mathbf{s}_i \in \mathbb{R}^{2d}; i = 1, \ldots, K\right\}$. Note that the i^{th} relation has a corresponding i^{th} instance. We encourage mutual interaction between instances and relations to help the model grasp the semantic correlation in depth. Specifically, in Figure 4.1, after acquiring representations of the instances and relations, we consider each i^{th} pair $(\mathbf{s}_i,\, \mathbf{r}_i)$ in turn. From the instance $\mathbf{s}_i$, we first compute its similarity to all K relations using the dot product operation and then use softmax to obtain a probability distribution over the K relations as follows:

$$z_{ij} = \frac{\exp\left(\mathbf{s}_i \cdot \mathbf{r}_j\right)}{\sum_{k=1}^{K} \exp\left(\mathbf{s}_i \cdot \mathbf{r}_k\right)} \tag{4.1}$$

where z_{ij} is the estimated probability for the j^{th} relation of the i^{th} instance. Let $\mathbf{z}_i = [z_{i1}, z_{i2}, \ldots, z_{iK}]$ denote the probability distribution of the i^{th} instance, which sums up to 1. The cross-entropy loss is calculated as follows:

$$\mathcal{L}_{CE} = -\log\left(z_{ii}\right), \tag{4.2}$$

where the i^{th} instance has a corresponding ground-truth i^{th} relation.

Similarly, we consider the interaction from each relation to instances. In Figure 4.1, from the relation $\mathbf{r}_i$ of the i^{th} pair $(\mathbf{s}_i,\, \mathbf{r}_i)$, we compute its similarity to all K instances using the dot product operation and then use softmax to obtain a probability distribution over the K instances as follows:

$$u_{ij} = \frac{\exp\left(\mathbf{r}_i \cdot \mathbf{s}_j\right)}{\sum_{k=1}^{K} \exp\left(\mathbf{r}_i \cdot \mathbf{s}_k\right)} \tag{4.3}$$

where u_{ij} is the estimated probability for the j^{th} instance of the i^{th} relation. Let $\mathbf{u}_i = [u_{i1}, u_{i2}, \ldots, u_{iK}]$ denote the probability distribution, which sums up to 1.

By considering the two-way interaction between instances and relations, for each i^{th} pair ($\mathbf{s}_i$, $\mathbf{r}_i$), we obtain two corresponding probability distributions ($\mathbf{z}_i$ and $\mathbf{u}_i$). These two distributions should be consistent to encourage the semantic correlation between the instance $\mathbf{s}_i$ and the relation $\mathbf{r}_i$. We then use the Kullback-Leibler (KL) divergence loss to supervise this consistency. Here, we deliberately use $D_{KL}\left(\mathbf{u}_i \| \mathbf{z}_i\right)$, instead of $D_{KL}\left(\mathbf{z}_i \| \mathbf{u}_i\right)$ or Jensen-Shannon divergence $D_{JS}\left(\mathbf{u}_i \| \mathbf{z}_i\right)$[1].

Because the natural language is highly flexible, a relation can be expressed using different textual patterns surrounding two entities in instances. For example, the relation "*per:employee_of*" can be reflected via patterns such as "worked for", "founded and headed", and "the CEO of". Thus, using the loss $D_{KL}\left(\mathbf{u}_i \| \mathbf{z}_i\right)$, which promotes $\mathbf{u}_i$ to be similar to $\mathbf{z}_i$, constrains the consistency of the two distributions and further encourages the model to learn richer and more diverse relation representations according to instances. The loss $D_{KL}\left(\mathbf{u}_i \| \mathbf{z}_i\right)$ is formulated:

$$\mathcal{L}_{KL} = D_{KL}\left(\mathbf{u}_i \| \mathbf{z}_i\right) = -\sum_{k=1}^{K} u_{ik} \log \frac{z_{ik}}{u_{ik}} \tag{4.4}$$

The final objective function of the model is defined as follows:

$$\mathcal{L} = \mathcal{L}_{CE} + \alpha \cdot \mathcal{L}_{KL} \tag{4.5}$$

where α is a hyperparameter that balances these two terms. Note that, for each training mini-batch that includes K different relations and K corresponding instances, we accumulate the losses of all these K pairs following the above formulation before using back-propagation in training.

[1]We tested with $D_{KL}\left(\mathbf{u}_i \| \mathbf{z}_i\right)$, $D_{KL}\left(\mathbf{z}_i \| \mathbf{u}_i\right)$, and $D_{JS}\left(\mathbf{u}_i \| \mathbf{z}_i\right)$ in our model. Using the loss $D_{KL}\left(\mathbf{u}_i \| \mathbf{z}_i\right)$ gave the best performance.

4.3 Experiments

4.3.1 Experimental Setup

Datasets. Following previous studies (Chen and Li, 2021; Gong and Eldardiry, 2021), we evaluate our model on two benchmark datasets: **FewRel** (Han et al., 2018) and **Wiki-ZSL** (Chen and Li, 2021). FewRel is a human-annotated balanced dataset comprising 80 relation types, each with 700 instances. Although FewRel was initially used for a few-shot learning task, it is also relevant for zero-shot learning, if the relation labels within training and test sets are disjoint.

In contrast, Wiki-ZSL originated from Wiki-KB (Sorokin and Gurevych, 2017) and is generated with distant supervision. From the Wiki-KB dataset, Chen and Li (2021) neglected instances with the relation "none". To ensure sufficient data instances for each relation in zero-shot learning, they filtered out relations that appeared less than 300 times. Finally, they obtained Wiki-ZSL, a subset of Wiki-KB. The statistics for Wiki-KB, Wiki-ZSL, and FewRel are shown in Table 4.1. Note that descriptions of all relations in Wiki-ZSL and FewRel are available from open-source Wikidata[2].

Zero-shot Settings. We follow the experimental settings of Chen and Li (2021) to enable the zero-shot relation extraction scenario. We randomly select m relations as *unseen* ones ($m = |\mathcal{Y}_u|$), thereby splitting the entire dataset into training and test sets; here, the test set includes all instances belonging to these m relations and the training set with all remaining instances. This ensures that these m relations are not in the training data such that $\mathcal{Y}_\mathcal{S} \cap \mathcal{Y}_\mathcal{U} = \emptyset$. Note that we repeat the experiment 5 times for 5 different random selections of m and report the average results. The evaluation metrics macro precision (P), macro recall (R), and macro F1-score ($F1$) are also used in this study, similar to previous studies.

Implementation Details. Our approach is implemented using PyTorch (Paszke et al., 2019) and all experiments are performed on 1 NVIDIA RTX A6000 GPU. We adopt the transformer library of Huggingface (Wolf et al., 2020) and use the

[2]https://www.wikidata.org/wiki/Wikidata:Main_Page

	#instances	#relations	avg. len.
Wiki-KB	$1,518,444$	354	23.82
Wiki-ZSL	$94,383$	113	24.85
FewRel	$56,000$	80	24.95

Table 4.1: Statistics of the datasets. "avg. len." stands for the average instance length.

Model	Wiki-ZSL			FewRel		
	Precision	*Recall*	*F1*	*Precision*	*Recall*	*F1*
CNN* (Zeng et al., 2014)	14.58	17.68	15.92	14.17	20.26	16.67
Bi-LSTM* (Zhang et al., 2015)	16.25	18.94	17.49	16.83	27.62	20.92
Att Bi-LSTM* (Zhou et al., 2016)	16.93	18.54	17.70	16.48	26.36	20.28
R-BERT* (Wu and He, 2019)	17.31	18.82	18.03	16.95	19.37	18.08
ESIM* (Chen et al., 2017)	27.31	29.62	28.42	29.15	31.59	30.32
CIM* (Rocktäschel et al., 2016)	29.17	30.58	29.86	31.83	33.06	32.43
ZS-BERT* (Chen and Li, 2021)	34.12	34.38	34.25	35.54	38.19	36.82
ZS-BERT† (Chen and Li, 2021)	38.22	33.70	35.72	38.96	37.35	38.06
ZS-SKA (Gong and Eldardiry, 2021)	41.03	40.12	38.13	45.34	51.67	47.02
Ours	**64.68**	**66.01**	**65.30**	**66.44**	**69.29**	**67.82**

Table 4.2: Results with $m = 15$ on Wiki-ZSL and FewRel. * indicates the results reported by Chen and Li (2021); † marks the results we reproduced using the official source code of Chen and Li (2021).

uncased model of BERT$_{base}$ as the encoder. The AdamW optimizer (Loshchilov and Hutter, 2019) is applied to minimize loss. For the hyperparameters, α is set to 1, and K is set to 5 on FewRel and Wiki-ZSL datasets. The maximum length of the instances is set to 128. The initial learning rate is $2e-5$. The number of sampled mini-batches is $40,000$. The hidden size, d is 768. The average runtime of our model's training and evaluation is 1.8 hours on Wiki-ZSL, whereas this number is 4.5 hours on FewRel.

4.3.2 Results and Analysis

Comparison to Baselines. The proposed approach is compared with the following baseline methods: **CNN** (Zeng et al., 2014), **Bi-LSTM** (Zhang et al., 2015), **Attention-based Bi-LSTM** (Zhou et al., 2016), **R-BERT** (Wu and He, 2019), **ESIM** (Chen et al., 2017), **CIM** (Rocktäschel et al., 2016), and **ZS-BERT** (Chen and Li, 2021). These baselines were reported by Chen and Li (2021). We further compare our model with the most state-of-the-art model, **ZS-SKA** by Gong and Eldardiry (2021).

Table 4.2 presents the experimental results for the Wiki-ZSL and FewRel datasets. Our approach significantly outperforms the strong baseline models by a significant margin, particularly for Wiki-ZSL. Specifically, our model improves the performance by 27.17 points and 20.8 points in $F1$-score on Wiki-ZSL and FewRel, respectively, compared to the state-of-the-art model ZS-SKA. The performance gain comes from the ability of our model to grasp the semantic correlation between instances and relations. Indeed, our model was entirely designed for this goal in three ways. (1) Our model obtains high-quality relation representations. (2) The strategic design of mini-batches is aimed at semantic correlation learning. (3) Our approach constrains the consistency of the two-way interaction between instances and relations. These aspects are discussed in detail in the following subsections.

Impact of Relation Representation. We investigate the role of the relation representation quality in affecting the ZSRE task's performance. Recall that, for Wiki-ZSL and FewRel datasets, both relation labels and descriptions are provided.

Chen and Li (2021) exploited only relation descriptions and input them into the fixed pre-trained Sentence-BERT (Reimers and Gurevych, 2019) to obtain relation representations. First, we attempt to follow this method to acquire such relation representations and use them in our model. Table 4.3 reports that our model achieves *F1*-score of 40.31 points and 42.52 points on Wiki-ZSL and FewRel, respectively. Using the same fixed relation representations, our model still achieves better performance in $F1$ score by 4.59 points and 4.46 points on Wiki-ZSL and FewRel, respectively, compared to ZS-BERT by Chen and Li (2021). However,

Input	Module	Wiki-ZSL			FewRel		
		Precision	*Recall*	*F1*	*Precision*	*Recall*	*F1*
Relation Description	Fixed Sentence-BERT Encoder	38.49	42.31	40.31	41.93	43.19	42.52
Relation Label	BERT-based Encoder	53.80	55.01	54.37	58.06	56.67	57.34
Relation Description		61.86	62.40	62.11	62.35	63.08	62.69
Label + Description		64.68	66.01	**65.30**	66.44	69.29	**67.82**

Table 4.3: Impact of the different relation representations in our model.

using such fixed relation representations causes overfitting during training. More severely, this hinders our model from grasping the semantic correlation between instances and relations effectively.

Therefore, we obtain relation representations via a learnable BERT-based encoder (Section 4.2.2) in our model. Although relation labels and descriptions are available, Chen and Li (2021) only used relation descriptions, while Gong and Eldardiry (2021) only exploited relation labels to create relation representations. Whereas relation labels provide concise and summary relation information, relation descriptions provide more detailed relation information. Intuitively, they complement each other to yield the best relation representations. We examine this intuition by feeding different inputs into the BERT-based encoder of our model to generate relation representations.

In Table 4.3, using only the relation label with the learnable BERT-based encoder, our model also achieves an impressive performance in $F1$ scores of 53.47 points and 57.34 points on Wiki-ZSL and FewRel, respectively. It significantly enhances $F1$ score by 14.06 points and 14.82 points on Wiki-ZSL and FewRel, compared to using the fixed relation representations in our model. This result proves the vital role of learning high-quality relation representations in solving ZSRE. Furthermore, we also consider using only relation descriptions via the BERT-based encoder in our model. Compared with only relation labels, using only relation descriptions achieves better performance and improves the $F1$ score by 7.74 and 5.35 points on Wiki-ZSL and FewRel, respectively. This may be reasonable because relation descriptions provide better relation representations with more detailed information.

Finally, using relation labels and descriptions to generate relation representations, our model achieves the best performance for Wiki-ZSL and FewRel. Specifically, this combination improves the $F1$ score by 3.19 points and 5.13 points on Wiki-ZSL and FewRel, respectively, than using only relation descriptions. This indicates that relation labels and relation descriptions complement each other to provide relation information more thoroughly, thereby acquiring the best relation representations leading to the best performance.

Impact of the Hyperparameter K. Figure 4.1 shows the prepared mini-batches to train our model, where each mini-batch has K different relations and K corresponding instances. Such designed mini-batches facilitate the model in grasping the semantic correlation between instances and relations. Thus, we inspect how the hyperparameter K affects the system performance.

The numbers of relations in the entire datasets Wiki-ZSL and FewRel are 113 and 80, respectively. To enable the ZSRE scenario, we randomly select $m = 15$ relations as *unseen* relations, thereby splitting each dataset into the training and test sets. Accordingly, the numbers of *seen* relations in training on Wiki-ZSL and FewRel are 98 and 65. Therefore, we try K with several values in [2, 98] on Wiki-ZSL and [2, 65] on FewRel in the training stage. At each value K, the reported $F1$ score is the average result obtained by repeating the experiment 5 times for 5 different random selections of m ($m = 15$) testing *unseen* relations.

Figure 4.2 shows the experimental results for the two test sets. Our model achieves the best performance with $K = 5$ on Wiki-ZSL and FewRel, whereas it obtains the worst performance with $K = 2$. Interestingly, using the largest value K (i.e., $K = 98$ on Wiki-ZSL and $K = 65$ on FewRel) does not give the best performance. Conversely, compared to the best performance with $K = 5$, using the largest value K significantly decreases the performance in $F1$ score on both the datasets by 13.89 points on Wiki-ZSL and 9.87 points on FewRel. Clearly, when using a substantial value K (e.g., $K = 98$ on Wiki-ZSL), our model can easily be distracted from fully grasping the interaction between a large number K of relations and K instances in each training mini-batch. It hinders the model from profoundly gripping the semantic correlation between instances and relations in the training phase, thereby causing a drop in performance in the testing stage. This also proves that selecting a relevant value K is essential to aid the model in effectively grasping the semantic correlation between instances and relations.

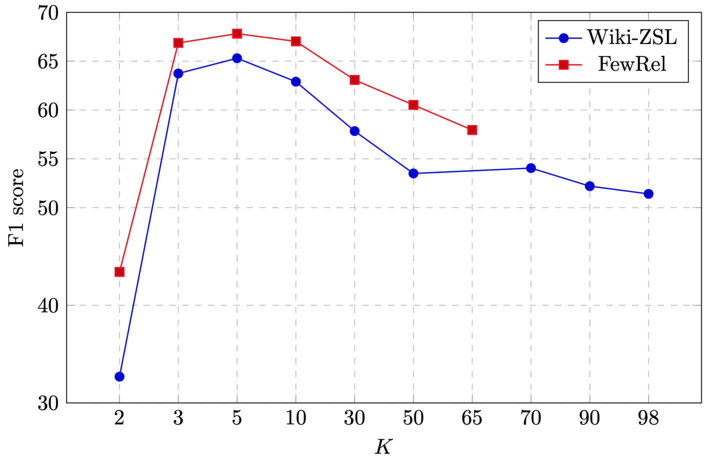

Figure 4.2: Impact of the hyperparameter K.

Our Model	Wiki-ZSL			FewRel		
	Precision	*Recall*	*F1*	*Precision*	*Recall*	*F1*
$\mathcal{L}_{CE} + \mathcal{L}_{KL}$	64.68	66.01	**65.30**	66.44	69.29	**67.82**
$\mathcal{L}_{CE}$	60.28	64.05	62.06	61.40	68.22	64.61

Table 4.4: Impact of using the loss $\mathcal{L}_{KL}$ (with $\alpha = 1$) in our model.

Meanwhile, previous studies (Chen and Li, 2021; Gong and Eldardiry, 2021) only simply compared each instance to the total training *seen* relations in learning the semantic relationship. By contrast, our method of controlling the K helps the model focus on thoroughly gripping the semantic correlation between instances and relations.

Impact of the Loss $\mathcal{L}_{KL}$. We use the final objective function (Equation 4.5) in the training stage to encourage the model to grasp the semantic interaction between instances and relations. As defined in Equation 4.5, the objective function comprises $\mathcal{L}_{CE}$ and $\mathcal{L}_{KL}$. While $\mathcal{L}_{CE}$ plays a significant role in learning the semantic correlation, $\mathcal{L}_{KL}$ also helps strengthen this goal by constraining two-way semantic distribution consistency. Examining the necessity of using $\mathcal{L}_{KL}$ in our model, we attempt to remove $\mathcal{L}_{KL}$ from the final objective function. The results

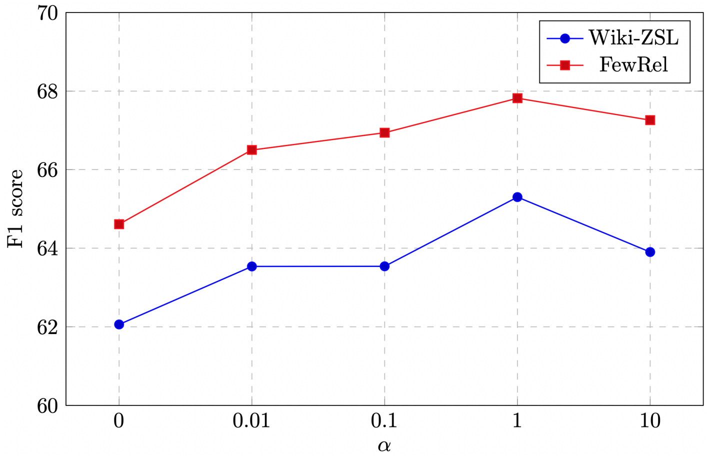

Figure 4.3: Impact of the hyperparameter α.

are presented in Table 4.4. Without using $\mathcal{L}_{KL}$, the system performance significantly decreases by 3.24 and 3.21 points in $F1$ score on Wiki-ZSL and FewRel, respectively. These results reaffirm the vital role of the loss $\mathcal{L}_{KL}$ in supervising our model to grasp the semantic correlation between instances and relations, thereby solving the ZSRE effectively.

We further investigate the sensitivity of α to the task performance by changing different values of α. As shown in Figure 4.3, our model achieves the best performance with $\alpha = 1$ on both Wiki-ZSL and FewRel datasets. Besides, when the value α is small (e.g., $\alpha = 0.01$) or quite large (e.g., $\alpha = 10$), it reduces the beneficial effect of using $\mathcal{L}_{KL}$ in improving the system performance.

Performance on Limited Labeled Data. We further examine the robustness of our model in solving the ZSRE under a limited labeled data scenario. As described in Table 4.1, FewRel is a human-annotated balanced dataset consisting of 80 relation types, each of which has 700 instances. First, we randomly split FewRel into training and test sets, where the training set includes 65 seen relations and the test set consists of 15 unseen relations. We then fix the test set and change the rate of the labeled data to train the models. Subsequently, the number of the seen relations of the training set is fine-tuned in [10, 65]. Note that the experiment is repeated 5 times for 5 different random data divisions, and we report the average results. We also run such experiments on ZS-BERT and compare it

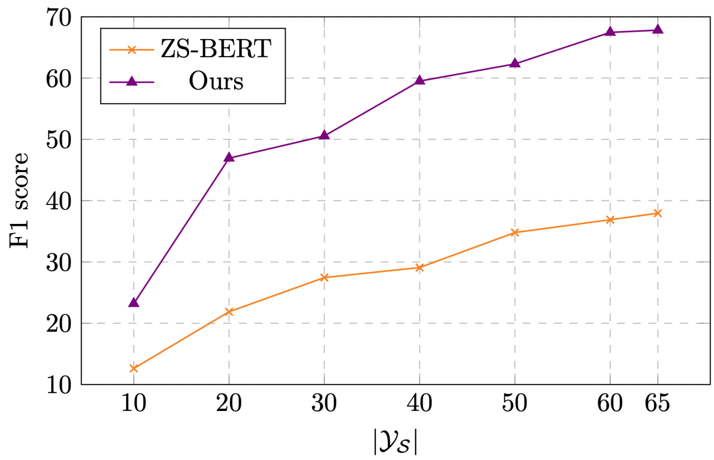

Figure 4.4: Performance on the limited labeled data.

with our model under similar limited training data conditions because only the official source code of ZS-BERT is available. Figure 4.4 shows the experimental results. Our model gains 2.0 times higher accuracy than ZS-BERT in $F1$ score in all limited data cases. This proves the robustness of our model in dealing with ZSRE under severely limited data conditions, which are popular in real-world scenarios.

Impact of Random Seed Sensitivity. All previously reported experimental results are the average results obtained by running the experiment 5 times with 5 random m selections ($m = 15$). We used the same fixed random seed in all of these experiments. Thus, we further check the sensitivity of our model to different random seeds for system performance.

We first split the entire Wiki-ZSL dataset into a training set and a test set, where the training set includes 98 seen relations and the test set consists of 15 unseen relations. Then, we try 5 different random seeds to train our model and report the average testing results. We repeat this process 3 times and report all the results in Table 4.5. Our model consistently outperforms ZS-BERT by a significant margin in the $F1$ score, all three times. More importantly, based on the standard deviations of the $F1$ scores, our model is more stable than ZS-BERT when training with different random seeds.

id	Model	*Precision*	*Recall*	*F1*
1	ZS-BERT	36.82	33.62	$\mathbf{35.12 \pm 1.88}$
	Ours	59.58	63.29	$\mathbf{61.34 \pm 1.65}$
2	ZS-BERT	46.28	41.33	$\mathbf{43.65 \pm 2.75}$
	Ours	75.59	77.71	$\mathbf{76.63 \pm 2.00}$
3	ZS-BERT	32.23	32.84	$\mathbf{32.50 \pm 2.41}$
	Ours	61.47	62.75	$\mathbf{62.10 \pm 1.90}$

Table 4.5: Impact of using different seeds to performance. The scores of ZS-BERT and our model are the average results of five runs with five different seeds. *F1* score is in the format of *mean* $\pm$ *standard deviation*.

4.4 Conclusion

This chapter presents a novel approach focusing entirely on enhancing the semantic correlation between instances and relations, which is key to solving ZSRE. Our approach concentrates on three major aspects to achieve this goal: learning high-quality relation representations, designing purposeful mini-batches, and binding two-way semantic distribution consistency. Extensive experiments on two benchmark datasets have demonstrated the effectiveness and robustness of our proposed model, particularly in limited training data scenarios. However, our approach is tested on the two benchmark datasets in the general domain and might not work well in specialized domains like the biomedical field. We plan to evaluate our method in such domains in future work.

Chapter 5

Improved Decomposition Strategy for Joint Entity and Relation Extraction

5.1 Introduction

The extraction of relational triplets is a critical and challenging task in natural language processing (NLP). Given an unstructured text, it aims to extract pairs of entities with semantic relations, in the form of (*head, relation, tail*). The relational triplets extraction has attracted considerable research effort as it plays a vital role in many NLP applications such as information extraction (Tran et al., 2021) and question answering (Hao et al., 2017). For example, in information extraction, given a biomedical text, it is expected to extract both the biomedical entities and their relations in the form of triplets such as ("*coronavirus*", "causes", "*respiratory infections*") and ("*tocilizumab*", "treats", "*cytokine release syndrome*").

Traditional pipeline works (Zelenko et al., 2003; Zhou et al., 2005; Chan and Roth, 2011) divide this task into two isolated subtasks: named-entity recognition (NER) (Vu et al., 2015) and relation classification (RC) (Tran et al., 2019). Specifically, they first recognize all the entities and then predict relations between the extracted entities. Such methods tend to suffer from error propagation and ignore the relevance between the two subtasks. To address these problems, subsequent studies proposed joint learning of entities and relations in a single model, including feature-based models (Yu and Lam, 2010; Li and Ji, 2014; Ren et al., 2017) and neural network-based models (Gupta et al., 2016; Katiyar and Cardie, 2017; Zeng et al., 2018; Fu et al., 2019; Yu et al., 2020).

One of the biggest challenges of this task is the *overlapping triplet problem*, which is expressed in two scenarios: *entity pair overlap* (**EPO**) and *single entity overlap* (**SEO**). Specifically, EPO occurs when triplets share the same entity pair but with different relations, such as: ("*Paris*", "Capital_of", "*France*"), ("*Paris*", "Located_in", "*France*"), and ("*Paris*", "Administrative_division_of", "*France*"), as in the sentence: "John Smiths lives and works in Paris, the capital and an administrative division of France". SEO occurs when two relational triplets share only one common entity, such as: ("*John Smiths*", "Work_in", "*Paris*") and ("*John Smiths*", "Live_in", "*France*").

Most previous works could not efficiently address the *overlapping triplet problem*. This problem directly challenges conventional sequence tagging schemes, in which each token represents only a single tag (Zheng et al., 2017). It also creates significant difficulties in traditional RC approaches, where an entity pair is supposed to hold at most one relation (Miwa and Bansal, 2016). Zeng et al. (2018) is among the first to solve the problem by proposing a sequence-to-sequence model with a copy mechanism. Fu et al. (2019) utilized a graph convolutional network to extract overlapping triplets. In contrast to the previous works, Yu et al. (2020) presented a unified sequence labeling framework based on a novel decomposition strategy. However, this method can only deal with the SEO triplets in the sample and fails to handle the EPO cases, as Yu et al. (2020) stated.

Specifically, Yu et al. (2020) decomposed the joint task into two subtasks: head-entity extraction and tail-entity relation extraction. The first task detects all head-entities, whereas the second one detects the corresponding tail-entities and target relations for a given head-entity. Although this method significantly outperforms previous methods, it suffers from two issues. **First**, to create relational triplets, it always detects head-entities first and then extracts the corresponding tail-entities and relations for each detected head entity. Thus, observably, if the first task fails to find a valid head-entity, the model will then miss all the related triplets containing this head-entity in the *head* role. **Second**, as Yu et al. (2020) stated, their model cannot solve the *overlapping triplet problem* in the

EPO scenario. For a given head-entity, the second task predicts only a single relation between the given head-entity and any corresponding tail-entity, even though this entity pair can hold multiple relations.

Therefore, we propose an improved decomposition strategy to overcome these two problems. For the **first issue**, we designed a more flexible strategy. We detect all entities first, and then, for each extracted entity, we identify it in each (*head* / *tail*) entity role and extract the corresponding (*tail-entities* / *head-entities*) and relations. For the **second issue**, we define a set of "*unified relation labels*" (*URLs*), each of which represents a unique (unordered) subset of the full set of original relations. By using these *URLs* in a multiclass classifier, our model can solve the EPO problem. In addition, a corresponding model framework is introduced to deploy our new strategy. The experimental results on both two benchmark datasets showed that our approach significantly outperformed the previous approach of Yu et al. (2020) as well as previous state-of-the-art approaches.

5.2 Methodology

In this section, we first introduce the decomposition strategy of Yu et al. (2020) and then present our new strategy. In addition, a corresponding model framework is proposed for deploying our decomposition strategy.

5.2.1 Tagging Scheme

Yu et al. (2020) decomposed the joint extraction task into two interrelated subtasks: Head-Entity (*HE*) extraction and Tail-Entity Relation (*TER*) extraction. The *HE* extraction task is modeled by two sequence labeling tasks, one for identifying the start position and the other for the end position of the head-entities, respectively. The entity type is also labeled simultaneously at the head-entity positions. Meanwhile, for each identified head-entity, the *TER* extraction task is also modeled by two sequence labeling tasks, one for detecting the start position and the other for detecting the end position of the corresponding tail-entities. As is done for the *HE* detection, the relation type between the given head-entity and its corresponding tail-entity is also labeled in each position.

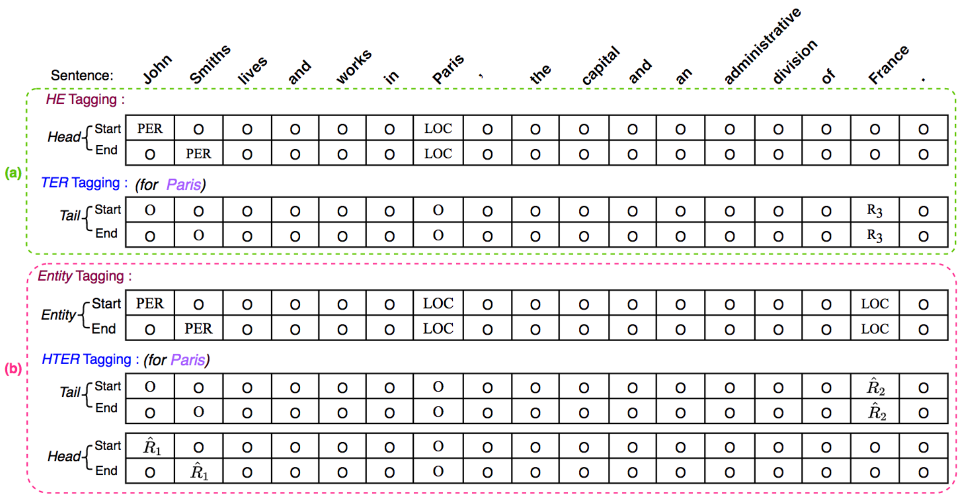

Figure 5.1: (a) Tagging scheme of Yu et al. (2020). (b) Our tagging scheme. *PER* and *LOC* stand for *PERSON* and *LOCATION*, respectively. *HE*, *TER*, and *HTER* stand for ***H**ead-**E**ntity*, ***T**ail-**E**ntity **R**elation*, and ***H**ead/**T**ail **E**ntity **R**elation*, respectively. The set of gold triplets is: {(*"John Smiths"*, R_1, *"Paris"*), (*"John Smiths"*, R_2, *"Paris"*), (*"John Smiths"*, R_1, *"France"*), (*"John Smiths"*, R_2, *"France"*), (*"Paris"*, R_3, *"France"*), (*"Paris"*, R_4, *"France"*), (*"Paris"*, R_5, *"France"*)}, where R_1: *"Live_in"*; R_2: *"Work_in"*; R_3: *"Capital_of"*; R_4: *"Located_in"*; and R_5: *"Administrative_division_of"*. In (b), $\hat{R}_1$ and $\hat{R}_2$ are *URLs*, where $\hat{R}_1$:{R_1, R_2} and $\hat{R}_2$:{R_3, R_4, R_5}.

Figure 5.1(a) illustrates an example of the above tagging scheme. From the input sample, the *HE* tagging detects the HEs: "*John Smiths*" and "*Paris*" because they are HEs in the set of gold triplets. Then, for the given HE "*Paris*", the *TER* tagging identifies the tail-entity "*France*" with the expected relation R_3 because of the gold triplet: (*"Paris"*, R_3, *"France"*). However, this tagging scheme suffers from two existing problems (as mentioned in Section 5.1) that hinder a further improvement of the system performance. We explain in detail how our new decomposition strategy can solve these two issues.

First, to obtain relational triplets in the form of (*head*, *relation*, *tail*), the model by Yu et al. (2020) always detects the HEs first and then extracts the corresponding tail-entities and relations for each detected HE. Following this strategy, if *HE* tagging fails to find a valid HE, the model will then miss all the related triplets.

For instance, in Figure 5.1(a), if "*Paris*" is not identified as a HE, the model will miss all gold triplets containing "*Paris*" in the *head* role. Meanwhile, it is not always easy to extract head entities first for all relations, especially when the relation types are diverse. To deal with this issue, we designed a new strategy, which is illustrated in Figure 5.1(b). This strategy allows our model not only to learn the probability distribution closer to the gold labels but also to increase the chances of extracting a valid triplet, which may be overlooked by the approach of Yu et al. (2020). Specifically, we first extract all entities without differentiating the *head*/*tail* role using the *Entity* tagging in our scheme. For each extracted entity, the *head/tail entity relation* (*HTER*) tagging considers it in each *head/tail* role and detects all corresponding *tail entities/head entities* and relation types, respectively. For example, in Figure 5.1(b), the *Entity* tagging detects the entities: "*John Smiths*", "*Paris*", and "*France*". Then, for the given entity "*Paris*", the *HTER* tagging considers it in the *head* role to identify the tail-entity "*France*" with the *unified relation label* (URL) $\hat{R}_2$, and also considers "*Paris*" in the *tail* role to recognize the HE "*John Smiths*" with the *URL* $\hat{R}_1$.

Second, the previous tagging scheme cannot solve the EPO problem. For instance, in Figure 5.1(a), the entity pair ("*Paris*", "*France*") holds multiple relations: R_3, R_4, and R_5. However, for the given HE "*Paris*", the *TER* tagging can predict only one of the original relations to the tail-entity "*France*", using a multi-class classifier of (N_R+1) classes, which include N_R original relations and one special class *No_relation*. To overcome this limitation, we propose two different solutions. First, in a natural way, we use a multi-label classifier to detect multiple relations (if any) between an ordered entity pair. With this solution, each tagging position in the *HTER* tagging can hold multiple original relation types (if any), instead of only a maximum of a single relation type (if any), as assumed by Yu et al. (2020). However, in practice, the maximum number of relation types co-occurring between an entity pair is often small[1]. For instance, the maximum

[1] There are two possible reasons for this phenomenon. First, because each relation type is usually associated with certain entity types (e.g., "*Live_in*" is between *Person* and *Location*), relation types co-occurring in the same entity pair are often required to share the same entity type pair. Second, in the joint extraction task, as each entity is often mentioned only once and the length of the input sample is not very long, they lead to the limited expressions of the possible relations of the same entity pair.

number of relation types for any entity pair is only 3 in both the NYT (Riedel et al., 2010) and the WebNLG (Gardent et al., 2017) datasets, while the total number of original relations on the NYT and WebNLG datasets is 24 and 216, respectively. Consequently, the sparse label problem on the relation types of the same entity pair can affect the system performance, especially in the WebNLG dataset. Therefore, we propose a second solution that uses a multi-class classifier with a set of *URLs* to deal with both the sparse label problem and the EPO problem. In essence, the purpose of using the created *URLs* is to transform the "multi-label classification task with a sparse label problem on the set of original relations" into the "multi-class classification task on the set *URLs*".

Using the training set $\boldsymbol{D}$ and a predefined threshold γ, following Algorithm 1, we create the set *URLs*. Specifically, first of all, for each ordered entity pair $\mathbf{p}$ in each sample in $\boldsymbol{D}$, the function $F(\mathbf{p})$ returns a single *URL* $\hat{R}$ that represents a unique (unordered) subset, where this subset includes all the existing original relations of the pair $\mathbf{p}$. We then count the frequency of each $\hat{R}$ on the entire $\boldsymbol{D}$ and only keep $\hat{R}$ when its frequency is greater than or equal to γ. With the obtained *URLs*, for each ordered entity pair *(head, tail)* in each sample in $\boldsymbol{D}$, we replace the full set of all existing original relations of this entity pair with a single corresponding *URL* in the set *URLs*. Conversely, we ignore relation sets if they do not match any corresponding *URLs* in the set *URLs*. Note that we only performed this label conversion for the training set, but not for the validation and test sets. With this procedure, any ordered entity pairs in any sentences in the training dataset $\boldsymbol{D}$ will now have only a single *URL* in the set *URLs* or have no relation. Finally, we train the model on the training dataset $\boldsymbol{D}$ with the *URLs* instead of with the set of original relations.

In Table 5.1, we provide a toy example for creating *URLs* using Algorithm 1 and for using them on the training set $\boldsymbol{D}$. Assume that the training set $\boldsymbol{D}$ includes two samples, where each sample has its gold relational triplets. By using Algorithm 1, we obtain the dictionary Q, which contains all the "*URLs*" along with their frequencies. With the predefined threshold γ (e.g.; γ=1), we obtain the set *URLs*: $\{\hat{R}_1, \hat{R}_2, \hat{R}_3, \hat{R}_4\}$. Then, using the created set *URLs*, for each entity pair in each sample, we replace all existing original relations of this pair with a single corresponding *URL* in the set *URLs* (if any). For instance, in Sample 2, the

Algorithm 1 Creation of a set of "*unified relation labels*"

Input: $\boldsymbol{D}$: training dataset; γ: a pre-defined threshold.

Output: *URLs*, the expected "*unified relation labels*" set.

Initialize an empty dictionary: $Q \leftarrow \{\}$
for each sample $\mathcal{X}$ in $\boldsymbol{D}$ **do**
    for each ordered entity pair $\mathbf{p}$ in $\mathcal{X}$ **do**
        $\hat{R} = F(\mathbf{p})$
        if $\hat{R} \neq \varnothing$ and $\hat{R}$ not in Q **then**
            $Q[\hat{R}] = 0$
        end if
        $Q[\hat{R}] = Q[\hat{R}] + 1$
    end for
end for
for each $\hat{R}$ in Q **do**
    if $Q[\hat{R}] \geq \gamma$ **then**
        Add $\hat{R}$ to the set *URLs*
    end if
end for
return *URLs*

ordered entity pair: ("*Alex*", "*Spain*") with the original relations: {"Work_in", "Place_of_birth", "Place_of_death"} will become: ("*Alex*", $\hat{R}_4$, "*Spain*"). Finally, our designed model will be trained on the training set $\boldsymbol{D}$ with the set *URLs*.

D	Sample 1	Harry works as an artist in Rome, the capital of Italy.
		("*Harry*", "Occupation", "*artist*"), ("*Harry*", "Work_in", "*Rome*"), ("*Harry*", "Work_in", "*Italy*"), ("*Rome*", "Capital_of", "*Italy*"), ("*Rome*", "Located_in", "*Italy*")
	Sample 2	Alex, a talented writer, was born and passed away in Spain, where he worked all his life.
		("*Alex*", "Occupation", "*writer*"), ("*Alex*", "Place_of_birth", "*Spain*"), ("*Alex*", "Place_of_death", "*Spain*"), ("*Alex*", "Work_in", "*Spain*")
Original Relations		"Occupation", "Work_in", "Capital_of", "Located_in", "Place_of_birth", "Place_of_death"
Unified Relation Labels		$\hat{R}_1$: {"Occupation"}, $\hat{R}_2$: {"Work_in"}, $\hat{R}_3$: {"Capital_of", "Located_in"}, $\hat{R}_4$: {"Place_of_birth", "Place_of_death", "Work_in"}.
Dict Q and the set *URLs*		Q = {$\hat{R}_1$: 2, $\hat{R}_2$: 2, $\hat{R}_3$: 1, $\hat{R}_4$: 1}. *URLs* = {$\hat{R}_1$, $\hat{R}_2$, $\hat{R}_3$, $\hat{R}_4$} when $\gamma = 1$.
D with the set *URLs*	Sample 1	Harry works as an artist in Rome, the capital of Italy.
		("*Harry*", $\hat{R}_1$, "*artist*"), ("*Harry*", $\hat{R}_2$, "*Rome*"), ("*Harry*", $\hat{R}_2$, "*Italy*"), ("*Rome*", $\hat{R}_3$, "*Italy*")
	Sample 2	Alex, a talented writer, was born and passed away in Spain, where he worked all his life.
		("*Alex*", $\hat{R}_1$, "*writer*"), ("*Alex*", $\hat{R}_4$, "*Spain*")

Table 5.1: A toy example of creating and using the set *URLs* on the training set ***D***.

5.2.2 Network Structure

Following our tagging scheme in Figure 5.1(b), we present our corresponding model framework in Figure 5.2. It consists of three main parts: *Encoding Layer*, *Entity Extractor*, and *HTER Extractor*.

Encoding Layer. Given a sample $X = \{x_1, x_2, ..., x_N\}$ with N tokens, we first utilize a bidirectional long short-term memory (BiLSTM) (Hochreiter and Schmidhuber, 1997) network to encode the contextualized representation for each token. The initial embedding $\mathbf{e}_i$ of each input token is concatenated by three parts: pre-trained word embedding, character-level word embedding generated by a convolutional neural network (CNN) on the character sequence of x_i, and a part-of-speech (POS) embedding. Then, the contextualized representation sequence $H = \{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_N\}$ is obtained as follows:

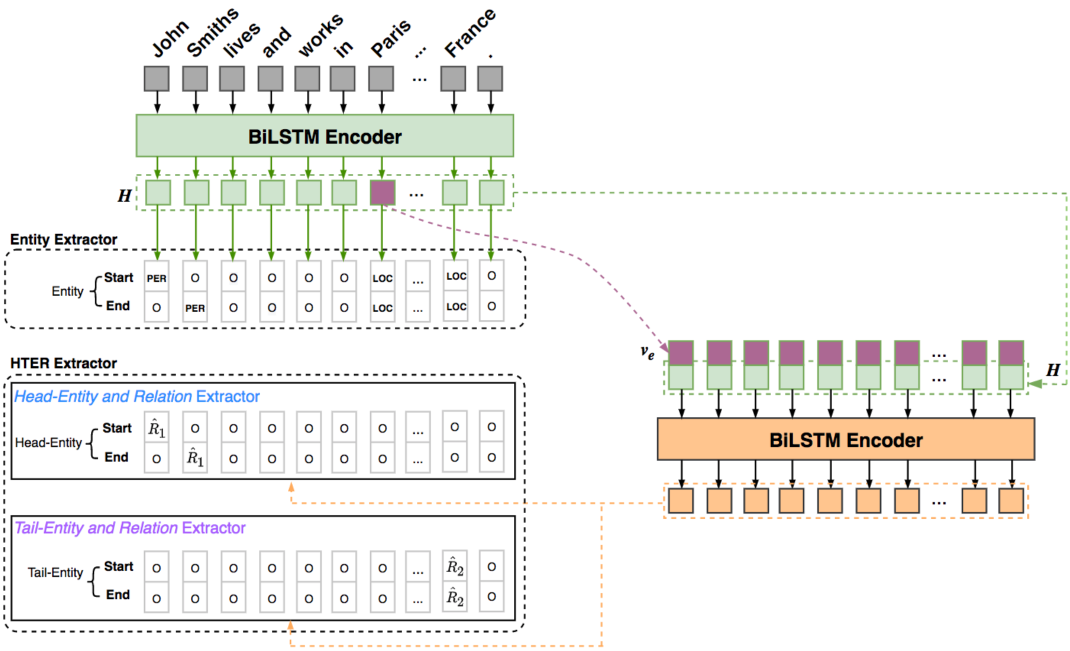

Figure 5.2: Our framework. We used the same input sample as in Figure 5.1. Here, the extracted entity "*Paris*" is entered into the HTER Extractor as prior knowledge. In the HTER Extractor, $\hat{R}_1$ and $\hat{R}_2$ are in the set *URLs* created using Algorithm 1, where $\hat{R}_1$: {"Live_in", "Work_in"}, $\hat{R}_2$:{"Capital_of", "Located_in", "Administrative_division_of"}. Note that the HTER Extractor was trained with the set *URLs*, instead of with the set of original relations.

$$\mathbf{h}_i = \left[\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i\right], \tag{5.1}$$

$$\overrightarrow{\mathbf{h}}_i = \mathrm{LSTM}^f\left(\mathbf{e}_i, \overrightarrow{\mathbf{h}}_{i-1}\right), \overleftarrow{\mathbf{h}}_i = \mathrm{LSTM}^b\left(\mathbf{e}_i, \overleftarrow{\mathbf{h}}_{i+1}\right), \tag{5.2}$$

where LSTM^f and LSTM^b denote the forward and backward LSTM, respectively.

Entity Extractor. The Entity Extractor module aims to recognize the relevant entities in the input sample by directly decoding the output sequence H of the Encoding Layer. Specifically, it adopts two identical multiclass classifiers to detect the start and end positions of the entities with the corresponding entity type label. Formally, the detailed operations of the entity tagging on each token are as follows:

$$p_i^{start\_ent} = Softmax\left(\mathbf{W}_{start\_ent}\mathbf{h}_i + \mathbf{b}_{start\_ent}\right), \tag{5.3}$$

$$p_i^{end\_ent} = Softmax\left(\mathbf{W}_{end\_ent}\mathbf{h}_i + \mathbf{b}_{end\_ent}\right), \tag{5.4}$$

where $p_i^{start\_ent}$ and $p_i^{end\_ent}$ represent the probabilities of the entity type labels for the i^{th} token, which are considered as the start and end positions of an entity, respectively. In addition, $\mathbf{h}_i$ is the encoded representation, $\mathbf{W}_{(.)}$ represents the trainable weight, and $\mathbf{b}_{(.)}$ is the bias.

We define the training loss (to be minimized) of the Entity Extractor as the sum of the negative log probabilities of the true *start* and *end* tags, using the predicted distributions:

$$\mathcal{L}_E = -\frac{1}{N}\sum_{i=1}^{N}\left(\log P\left(y_i^{start\_ent} = \hat{y}_i^{start\_ent}\right) + \log P\left(y_i^{end\_ent} = \hat{y}_i^{end\_ent}\right)\right), \tag{5.5}$$

where $\hat{y}_i{}^{start\_ent}$ and $\hat{y}_i{}^{end\_ent}$ are the true *start* and *end* tags (gold labels) of the i^{th} word in the sample X, respectively, and N is the length of the sample X.

HTER Extractor. The *HTER Extractor* consists of two submodules: ***H**ead-**E**ntity **R**elation* (*HER*) extractor and ***T**ail-**E**ntity **R**elation* (*TER*) extractor. For each given entity, e.g., "Paris", it uses the *TER* to identify "Paris" in the *head* entity role and detect all the corresponding tail-entities and *URLs*, such as (*"**Paris**"*, $\hat{R}_2$, *"France"*), where $\hat{R}_2$:{"Capital_of", "Located_in", "Administrative_division_of"}. At the same time, the HTER Extractor utilizes the *HER* submodule to identify "Paris" in the *tail* entity role and detect all the corresponding head-entities and *URLs*, such as (*"John Smiths"*, $\hat{R}_1$, *"**Paris**"*), where $\hat{R}_1$:{"Live_in", "Work_in"}.

Specifically, from the output sequence H of the Encoding Layer, as an entity is often composed of multiple tokens, we create a span feature representation for the given entity. Following Ouchi et al. (2018), for the entity with start and end positions: j and k ($j \leq k$), we obtain the entity representation vector as follows:

$$\mathbf{v}_{ent} = [\mathbf{h}_j + \mathbf{h}_k; \mathbf{h}_j - \mathbf{h}_k], \tag{5.6}$$

$$\overline{\mathbf{x}}_i = [\mathbf{h}_i; \mathbf{v}_{ent}], \tag{5.7}$$

where i refers to the position of the i^{th} word in the input sample.

Because the information of a given entity is crucial for extracting related triplets, we therefore concatenate each token vector in the output sequence H and the given entity representation $\mathbf{v}_{ent}$. We take $\overline{\mathbf{X}} = \{\overline{\mathbf{x}}_1, \overline{\mathbf{x}}_2, \cdots, \overline{\mathbf{x}}_N\}$ as the input to another BiLSTM layer, to fuse each $\mathbf{h}_i$ and $\mathbf{v}_{ent}$ in a single vector $\overline{\mathbf{h}}_i$:

$$\overline{\mathbf{H}} = BiLSTM(\overline{\mathbf{X}}), \tag{5.8}$$

where $\overline{\mathbf{H}} = \left\{\overline{\mathbf{h}}_1, \overline{\mathbf{h}}_2, \cdots, \overline{\mathbf{h}}_N\right\}$. Then, the sequence $\overline{\mathbf{H}}$ is used as the same input to both *TER* and *HER* submodules. The *TER* submodule detects all the corresponding tail-entities and relations by directly decoding the sequence $\overline{\mathbf{H}}$. Specifically, it uses two identical multiclass classifiers to detect the start and end positions of the related tail-entities with the corresponding relation type. Thus, the detailed operations of the *tail* entity tagging with the relation type on each token are described as follows:

$$p_i^{start\_tail} = Softmax\left(\mathbf{W}_{start\_tail}\overline{\mathbf{h}}_i + \mathbf{b}_{start\_tail}\right), \tag{5.9}$$

$$p_i^{end\_tail} = Softmax\left(\mathbf{W}_{end\_tail}\overline{\mathbf{h}}_i + \mathbf{b}_{end\_tail}\right), \tag{5.10}$$

where $p_i^{start\_tail}$ and $p_i^{end\_tail}$ represent the probabilities of the relation labels for the i^{th} token, which are considered as the start and end positions of a *tail* entity in the input sample, respectively. Additionally, $\overline{\mathbf{h}}_i$ is the encoded representation, $\mathbf{W}_{(.)}$ represents the trainable weight, and $\mathbf{b}_{(.)}$ is the bias.

Similarly, the *HER* submodule utilizes two other identical multi-class classifiers to detect the start and end positions of the related head-entities with the corresponding relation type. Formally, the detailed operations of the *head* tagging on each token are as follows:

$$p_i^{start\_head} = Softmax\left(\mathbf{W}_{start\_head}\overline{\mathbf{h}}_i + \mathbf{b}_{start\_head}\right), \tag{5.11}$$

$$p_i^{end\_head} = Softmax\left(\mathbf{W}_{end\_head}\overline{\mathbf{h}}_i + \mathbf{b}_{end\_head}\right). \tag{5.12}$$

Therefore, we have the loss function of each submodule in the *HTER Extractor* as follows:

$$\mathcal{L}_{TER} = -\frac{1}{N}\sum_{i=1}^{N}\left(\log P\left(y_i^{start\_tail} = \hat{y}_i^{start\_tail}\right) + \log P\left(y_i^{end\_tail} = \hat{y}_i^{end\_tail}\right)\right), \tag{5.13}$$

$$\mathcal{L}_{HER} = -\frac{1}{N}\sum_{i=1}^{N}\left(\log P\left(y_i^{start\_head} = \hat{y}_i^{start\_head}\right) + \log P\left(y_i^{end\_head} = \hat{y}_i^{end\_head}\right)\right), \tag{5.14}$$

where N is the length of the input sample; $\hat{y}_i{}^{start\_tail}$ and $\hat{y}_i{}^{end\_tail}$ in Equation 5.13 are the true *start* and *end* relation tags of the i^{th} word for annotating the related tail entities, respectively, and $\hat{y}_i{}^{start\_head}$ and $\hat{y}_i{}^{end\_head}$ in Equation 5.14 are the true *start* and *end* relation tags of the i^{th} word for annotating the related head entities.

Joint Learning. To boost the interaction between the *Entity Extractor* and the *HTER Extractor*, we combine their loss functions to form the entire loss objective of our model:

$$\mathcal{L}(\theta) = \alpha * \mathcal{L}_E + (\mathcal{L}_{TER} + \mathcal{L}_{HER}), \tag{5.15}$$

where the hyper-parameter α is fine-tuned in the range (0, 1]. Then, we train the model by minimizing $\mathcal{L}(\theta)$ through the Adam stochastic gradient descent (Kingma and Ba, 2015) over shuffled mini-batches. Note that the *HTER Extractor* is trained with the gold set *URLs*, which are created using Algorithm 1, instead of with the set of original relations.

Inference. In the testing phase, the triplets can be easily inferred on the basis of the two modules. Specifically, for each input sample, we first extract the entities by using the *Entity Extractor* module. Note that entities extracted by this module will not be considered as an additional constraint on the output of the other module. Then, for each detected entity, we utilize the *HTER Extractor* to consider it in the *head*/*tail* roles and extract all the relational triplets involving this entity. For example, from the input sample in Figure 5.2, the *Entity Extractor* is expected to detect the entities: "John Smiths", "Paris", and "France". Then, for each extracted entity, e.g., "Paris", the *HTER Extractor* uses its two submodules (*HER* and *TER*) to extract all relational triplets containing "Paris". Specifically, the *TER* submodule identifies "Paris" in the *head* role and extracts: ("***Paris***", $\hat{R}_2$, "*France*"). Meanwhile, the *HER* submodule considers "Paris" in the *tail* role and extracts: ("*John Smiths*", $\hat{R}_1$, "***Paris***").

Note that the relation types in the triplets extracted by both the *HER* and *TER* submodules belong to the set *URLs* because they are trained with this set. Thus, we need to transform these relations into the original relations by breaking them down into the original relations and creating the corresponding triplets. In the above example, for the given entity "*Paris*", the *TER* submodule extracts the triplets {("***Paris***", $\hat{R}_2$, "*France*")}. In addition, as shown in Figure 5.1, $\hat{R}_2$ represents for the subset {"Capital_of", "Located_in", "Administrative_division_of"}. Therefore, we obtain the final triplets from the *TER* submodule for "*Paris*": {("***Paris***", "Capital_of", "*France*"), ("***Paris***", "Located_in", "*France*"), ("***Paris***", "Administrative_division_of", "*France*")}. Similarly, we also obtain the triplets from the *HER* submodule for "*Paris*": {("*John Smiths*", "Live_in", "***Paris***"), ("*John Smiths*", "Work_in", "***Paris***")}. Finally, we combine the outputs from both submodules by keeping all the extracted triplets, but removing the duplicates (if any) for each input sample.

5.3 Experiments

5.3.1 Experimental Settings

Datasets and Evaluation Metrics. Following the previous work (Dai et al., 2019; Yu et al., 2020), we evaluated our approach on two widely used datasets: **NYT** (Riedel et al., 2010) and **WebNLG** (Gardent et al., 2017). To further study the capability of our approach to extract overlapping and multiple relations, we also split the test set into three categories: *Normal*, *EPO*, and *SEO*. A sample belongs to *Normal* if none of its triplets overlaps, whereas it belongs to *EPO* if

Dataset	Train	Valid	Test	Category			No. of Relations
				Normal	SEO	EPO	
NYT	$56,195$	$5,000$	$5,000$	$3,266$	$1,297$	978	24
WebNLG	$5,019$	500	703	216	457	26	216

Table 5.2: Statistics of the two datasets. The number of samples in the test set that belongs to each category, is also reported. Note that a sample can belong to both the *SEO* and *EPO* categories. In addition, the relation number of the WebNLG was miswritten as 246, as in (Fu et al., 2019; Yu et al., 2020), which is the total number of relations in the original WebNLG dataset instead of the number of the subsets they used. We recounted and provided the correct number.

some of its triplets share the same entity pair. In addition, a sample belongs to *SEO* if some of its triplets share only one common entity. The statistics of the two datasets are given in Table 5.2.

We report the standard micro precision, recall, and F1-score, as in line with recent studies. Specifically, a predicted triplet is correct if and only if its relation type and its two corresponding entities are all the same as those in the gold standard annotation. The results of the test set were reported when the development set achieved the best result.

Implementation Details. We implemented the neural networks using the PyTorch library[2]. Batch padding was applied to pad the lengths of all tokens to make them equal to the maximum length in each batch. The mini-batch training size was set to 64, which was selected from the set: [32, 50, 64].

We used the 300-dimensional GloVe (Pennington et al., 2014) to initialize the word embeddings. Each word representation was concatenated by three parts: pre-trained GloVe embedding, character-based word representation by running a CNN on the character sequence of the word, and POS embedding. The POS, character, and position embeddings were randomly initialized with 30 dimensions (selected from the set: [30, 40, 50]). The filter size of the CNN was set to 3 from

[2]PyTorch is an open-source software library for machine intelligence: https://pytorch.org/

the set: [3, 4, 5], and the number of filters was 50 from the set: [30, 40, 50]. Thus, the representation of each word had a dimensionality of 380 (as the input of the BiLSTM layer). For the BiLSTM layer, the hidden vector size was set to 200 from the set: [150, 200]. The *Adam* optimizer (Kingma and Ba, 2015) with a learning rate 0.0001 from the set: [0.0001, 0.00001] was employed for training. Dropout was applied to word embeddings and hidden states at a rate of 0.4 from the set: [0.3, 0.4]. We also set the gradient clip-norm to 5 to prevent the gradient explosion problem. The threshold γ was set to 11 for the NYT training set and to 7 for the WebNLG training set. In addition, the value of α in the final loss function (Equation 5.15) was set to 0.3 on the NYT and to 0.2 on the WebNLG, where α was in the range (0, 1]. We trained the model for 100 epochs on both datasets. Hyperparameters were tuned on the development set. All experiments were run on a Tesla V100 graphics card in an Ubuntu-based computer system.

5.3.2 Experimental Results and Analyses

Comparison Models. For comparison, we employed the following models as baselines:

- **NovelTagging** (Zheng et al., 2017): The first model to introduce a novel tagging scheme that transforms the joint extraction task into a sequence labeling problem.
- **MultiDecoder** (Zeng et al., 2018): A seq2seq model with a copy mechanism that converts the joint extraction task to a sequence-to-sequence problem.
- **MultiHead** (Bekoulis et al., 2018): A joint neural model that performs entity recognition and relation extraction simultaneously.
- **GraphRel** (Fu et al., 2019): An end-to-end relation extraction model that uses GCNs to jointly learn named entities and relations.
- **OrderRL** (Zeng et al., 2019): A sequence-to-sequence model with reinforcement learning that takes the extraction order into consideration.

- **ETL-Span** (Yu et al., 2020): A sequence labeling framework based on a novel decomposition strategy that has achieved a notable performance; however, its decomposition strategy still cannot solve the EPO problem, as the authors stated.

Main Results. Table 5.3 shows the results of our models against those of other baseline methods on both the NYT and WebNLG datasets. First, ETL-Span (Yu et al., 2020) with a decomposition strategy significantly outperformed the previous works by a wide margin. However, because this approach cannot solve the EPO problem as Yu et al. (2020) stated, further improvement of the system performance is hindered. Meanwhile, our model framework with a new decomposition strategy overcomes the existing problems of the model of Yu et al. (2020) and substantially boosts the system performance. Specifically, our approach improved the F1-score by 7.1 points on the NYT and by 2.9 points on the WebNLG, compared with the results of Yu et al. (2020).

Model	**NYT**			**WebNLG**		
	Precision	*Recall*	*F1*	*Precision*	*Recall*	*F1*
NovelTagging (Zheng et al., 2017)	32.8	30.6	31.7	52.5	19.3	28.3
MultiDecoder (Zeng et al., 2018)	61.0	56.6	58.7	37.7	36.4	37.1
MultiHead (Bekoulis et al., 2018)	60.7	58.6	59.6	57.5	54.1	55.7
GraphRel (Fu et al., 2019)	63.9	60.0	61.9	44.7	41.1	42.9
OrderRL (Zeng et al., 2019)	77.9	67.2	72.1	63.3	59.9	61.6
ETL-Span (Yu et al., 2020)	85.5	71.7	78.0	84.3	82.0	83.1
Ours	82.2	88.2	**85.1**	84.3	87.7	**86.0**

Table 5.3: Main results of the performances of the compared models on the NYT and WebNLG.

Analysis of Our Decomposition Strategy. To gain more insight into the improvement of our decomposition strategy in our model (in Figure 5.2), we conducted further experiments, as reported in Table 5.4. We also reproduced the results of ETL-Span (Yu et al., 2020).

Model	NYT			WebNLG		
	Precision	*Recall*	*F1*	*Precision*	*Recall*	*F1*
ETL-Span*	84.4	72.2	77.8	84.5	81.6	83.0
(**a**) Ours (multiclass)	81.4	76.6	78.9	83.3	86.9	85.1
(**b**) Ours (multilabel)	81.3	83.7	82.4	81.4	85.3	83.3
(**c**) Ours (multiclass + *URLs*)	82.2	88.2	**85.1**	84.3	87.7	**86.0**

Table 5.4: Analysis of the performance of our framework on the test sets. The * marks the results that we reproduced. The *URLs* are the set of "*unified relation labels*" created using Algorithm 1.

First, for case (**a**) in Table 5.4, we considered our model without using the set *URLs*. Specifically, we used multiclass classifiers on the set of original relations in the two submodules of the module *HTER Extractor*. Compared with the model of Yu et al. (2020), our model achieved gains of 1.1 points and 2.1 points in the $F1$-score on the NYT and WebNLG, respectively, using only the set of original relations. The model of Yu et al. (2020) is too strict in regard to the order it obtains the elements of each triplet, as it always detects the head entities first and then extracts the corresponding tail entities and relations for the given HE. Consequently, it will miss all triplets related to an omitted valid HE. Meanwhile, it is not always easy to extract head entities first for all relations, as in some cases it might be easier to detect the tail entities first before the head entities. Thus, our flexible approach overcomes this problem and significantly improves the recall. Note that our approach achieved a better improvement in the $F1$-score on the WebNLG than that on the NYT. One possible reason is that, because the number of relation types in the WebNLG (216 types) is much larger than that in the NYT (only 24 types), it increased the probability of relations where it was easier to detect the tail entities first before the head entities.

Second, as our multiclass model in case (**a**) cannot solve the EPO problem, we considered the first solution. Specifically, in case (**b**), we used multilabel classifiers, instead of multiclass classifiers, on the set of original relations in the two submodules of the *HTER Extractor*. With this solution, each tagging position in the *HTER Extractor* can hold multiple original relation types. Thus, we can extract multiple relations (if any) of the same entity pair. By doing this, compared to case (**a**), our system achieved a gain of 3.5 points in the $F1$-score on the NYT, whereas it showed a decreased of 1.8 points in the $F1$-score on the WebNLG. We observed that the main difference between the NYT and WebNLG might have led to this result. Specifically, the number of original relations in the WebNLG (216 types) is much larger than that in the NYT (24 types), although the maximum number of relations of the same entity pair is 3 on both of these training sets. Consequently, the sparse label problem of the multilabel classification on the same entity pair is more severe in the WebNLG than in the NYT. Therefore, it considerably affected the system performance on the WebNLG. Meanwhile, although this problem is less severe in the NYT than in the WebNLG, it also hinders the further improvement of the system performance.

Finally, as our model suffers from the sparse label problem for multilabel classification of the same entity pair in case (**b**), we considered the second solution to solve the EPO problem. Specifically, in case (**c**), because a multiclass classification can alleviate the sparse label problem, we used multiclass classifiers with the *URLs* created using Algorithm 1 in the *HTER Extractor*. Interestingly, by using this simple solution, we achieved the highest system performance for both the NYT and WebNLG. Compared with case (**a**), the solution increased the $F1$-score by 6.2 points and 0.9 points on the NYT and WebNLG, respectively. It is worth mentioning that the improvement gain on the NYT was significantly larger than that on the WebNLG. One possible reason is that the EPO problem on the NYT is more serious than that on the WebNLG. In Table 5.2, the number of samples belonging to the EPO category in the NYT test set is 978 (19.6%), whereas it is only 26 (3.7%) in the WebNLG test set.

Compared with the ETL-Span model by Yu et al. (2020), in Table 5.4, our best model (case (**c**)) achieved a significant improvement of the system performance with an increase in the $F1$-score by 7.3 points and 3.0 points on the NYT and WebNLG test sets, respectively. In addition, on the NYT test set, compared with the ETL-Span model, although our best model boosted the recall significantly by 16 points, the precision decreased by 2 points. One possible reason for the decrease in the precision is that our model tries to train all three parts (i.e., *Entity Extractor* and the two submodules: *TER* and *HER*) effectively at the

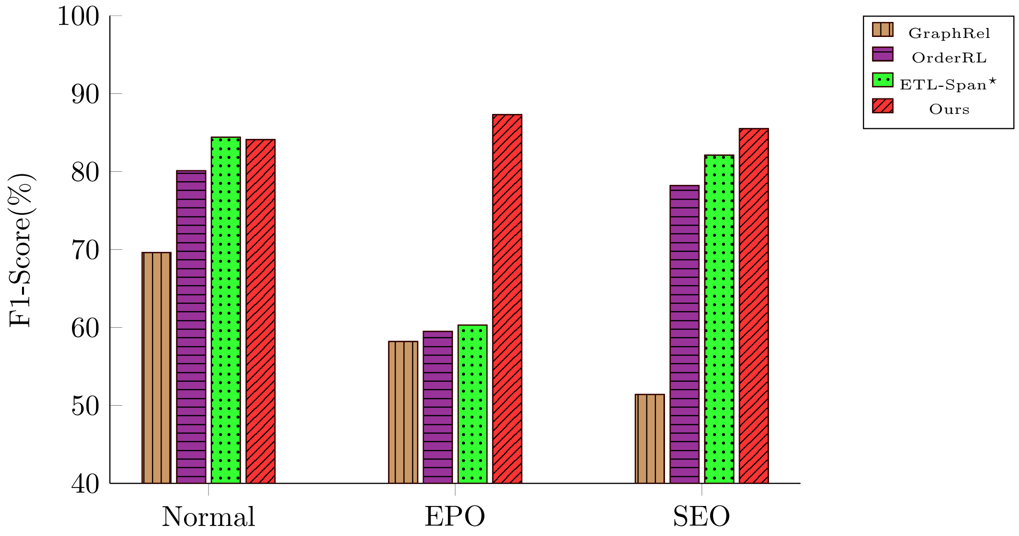

Figure 5.3: F1-score of extracting relational triplets from samples in three different categories on the NYT test set.

same time, which might be more challenging than training only two elements simultaneously (i.e., the *HE Extractor* and *TER Extractor*), as in the ETL-Span model of Yu et al. (2020). In future work, we plan to design model architectures more effectively, to obtain a satisfactory level of not only the recall measure but also the precision measure, thereby further improving the $F1$-score.

Analysis of Different Sample Types. To verify the capability of our model to extract multiple triplets, we followed the procedure in (Zeng et al., 2018; Fu et al., 2019) and conducted further experiments on the NYT test set. Specifically, we first split the samples in this test set into three categories: *Normal*, *EPO*, and *SEO*, and then we investigated the performance of each category.

The results are shown in Figure 5.3. It can be seen from the figure that the performance improvement in our model mainly comes from its ability to deal with the EPO and SEO problems more effectively. Compared with the model of Yu et al. (2020), our model achieved competitive performance in all the three categories. In addition, we paid special attention to the performance differences between our approach and that of Yu et al. (2020). Notably, on the NYT test

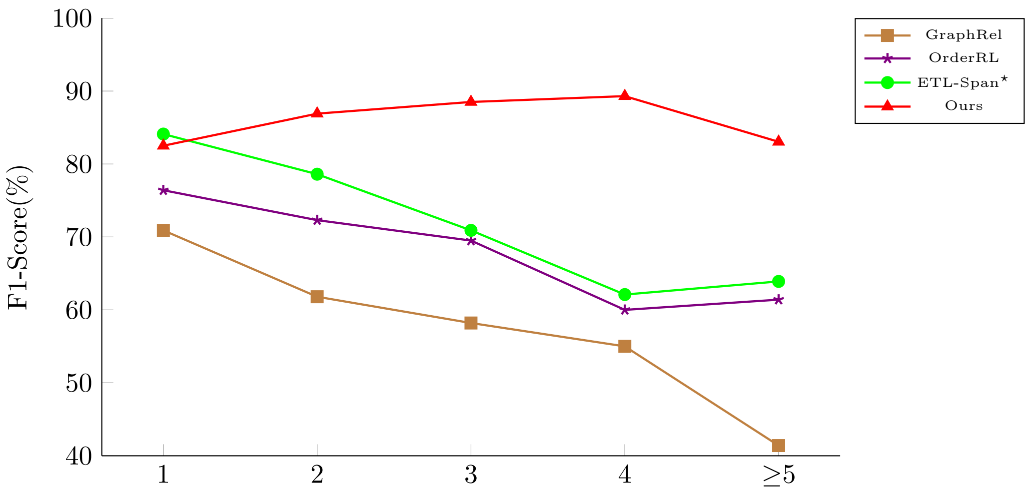

Figure 5.4: F1-scores obtained after extracting relational triplets from samples with different numbers of triplets on the NYT test set.

set, our approach boosted the $F1$-score significantly in the EPO problem by 27.0 points, whereas that of Yu et al. (2020) cannot solve this problem. In addition, as the strategy of Yu et al. (2020) strictly constrains the detection of the entities to the head first, if it fails to find a valid HE, it will then miss the related triplets. It will be more serious if this HE attends many different triplets in the *head* entity role (a case of SEO). Therefore, our flexible approach deals with this issue and substantially improves the $F1$-score by 3.4 points in the SEO problem.

We also compared the ability of the models to extract multiple triplets in a sample. Specifically, we divided the samples of the NYT test set into five categories, where each category contains samples that have 1, 2, 3, 4, or ≥ 5 triplets, respectively. The results are shown in Figure 5.4. It can be seen from the figure that our approach achieved a significant improvement in extracting multiple triplets compared with the other models. In particular, our model showed a more stable performance when the number of triplets in the sample increased. These results show that our approach is effective in dealing with the multi-relation extraction task.

5.3.3 Case Study

To gain more insight into the effectiveness of our model in overcoming the existing disadvantages in the approach of Yu et al. (2020), we analyzed the prediction outputs of both models on a few samples of the NYT and WebNLG test sets and these are shown in Tables 5.5 and 5.6, respectively.

Dealing With the EPO Problem. In Table 5.5, we show two examples from the NYT test set and compare the predicted triplets of the model of Yu et al. (2020) with those of our model.

Sample 1	Anti-Ethiopia riots erupted in Mogadishu, the capital of Somalia, on Friday, while masked gunmen emerged for the first time on the streets, a day after Ethiopian-backed troops captured the city from Islamist forces.
Yu et al. (2020)	(“*Somalia*”, “/location/location/contains”, “*Mogadishu*”)
Our model	(“*Somalia*”, “/location/country/capital”, “*Mogadishu*”) (“*Somalia*”,“/location/location/contains”, “*Mogadishu*”)
Ground Truth	(“*Somalia*”, “/location/country/capital”, “*Mogadishu*”) (“*Somalia*”, “/location/location/contains”, “*Mogadishu*”)
Sample 2	Though officials in Addis Ababa, Ethiopia’s capital, have said their troops should not enter downtown Mogadishu, many are camped in the former American Embassy, a decrepit building that was closed more than 15 years ago after American soldiers suffered a humiliating defeat at the hands of warlords.
Yu et al. (2020)	(“*Ethiopia*”, “/location/country/capital”, “*Mogadishu*”) (“*Ethiopia*”, “/location/location/contains”, “*Addis Ababa*”) (“*Addis Ababa*”, “/location/administrative_division/country”, “*Ethiopia*”)
Our model	(“*Ethiopia*”, “/location/country/capital”, “*Addis Ababa*”) (“*Ethiopia*”, “/location/country/administrative_divisions”, “*Addis Ababa*”) (“*Ethiopia*”, “/location/location/contains”, “*Addis Ababa*”) (“*Addis Ababa*”, “/location/administrative_division/country”, “*Ethiopia*”)
Ground Truth	(“*Ethiopia*”, “/location/country/capital”, “*Addis Ababa*”) (“*Ethiopia*”, “/location/country/administrative_divisions”, “*Addis Ababa*”) (“*Ethiopia*”, “/location/location/contains”, “*Addis Ababa*”) (“*Addis Ababa*”, “/location/administrative_division/country”, “*Ethiopia*”)

Table 5.5: Prediction outputs on two samples from the NYT test set.

As mentioned earlier, the model of Yu et al. (2020) cannot solve the EPO problem. For any entity pairs, their model only predicts a single relation, although an entity pair can have multiple relations. For instance, in Sample 1, the ordered entity pair (“*Somalia*”, “*Mogadishu*”) has two relations: “/location/country/capital” and “/location/location/contains”. However, the model of Yu et al. (2020) extracted only a single relation and created the triplet: (“*Somalia*”, “/location/location/contains”, “*Mogadishu*”). Similarly, in Sample 2, although the ordered entity pair (“*Ethiopia*”, “*Addis Ababa*”) has three relations: “/location/country/capital”, “/location/country/administrative_divisions”, and “/location/location/contains”, their model predicted only the relation “/location/location/contains” for this pair. Thus, the more serious the EPO problem is, the more degraded the system performance becomes. Meanwhile, our model overcomes this disadvantage and effectively solves the EPO problem. For both samples above, our model fully detected all possible relations for the pair (“*Somalia*”, “*Mogadishu*”) in Sample 1 and the pair (“*Ethiopia*”, “*Addis Ababa*”) in Sample 2.

Effect of the “*Exhaustive Search*” Strategy. As shown in Figure 5.2, our model uses the *Entity Extractor* to detect all entities first. Then, for each detected entity, the *HTER Extractor* utilizes its two submodules to identify the entity in each *head*/*tail* role and extracts all the corresponding *tail entities*/*head entities* and relations. The final output of our model is always obtained by combining the results of the two submodules without any duplicate triplets. In essence, this approach can be considered as an “*exhaustive search*” strategy that aims to increase the chances of extracting a valid triplet that may be overlooked by the approach of Yu et al. (2020). Therefore, in Table 5.6, we compare the prediction outputs of both approaches on three samples from the WebNLG test set.

Sample 3		The Athens International Airport serves the city of Athens , in Greece where Alexis Tsipras is the leader.
Yu et al. (2020)		(“*Athens*”, “country”, “*Greece*”) (“*Greece*”, “leaderName”, “*Alexis Tsipras*”)
Our model	*HER+URLs*	(“*Athens*”, “country”, “*Greece*”) (“*Greece*”, “leaderName”, “*Alexis Tsipras*”)
	TER+URLs	(“*Athens*”, “country”, “*Greece*”) (“*Greece*”, “leaderName”, “*Alexis Tsipras*”) (“*Athens International Airport*”, “cityServed”, “*Athens*”)
Ground Truth		(“*Athens*”, “country”, “*Greece*”) (“*Greece*”, “leaderName”, “*Alexis Tsipras*”) (“*Athens International Airport*”, “cityServed”, “*Athens*”)
Sample 4		Faber and Faber are the publishers of The Secret Scripture, a sequel to A Long Long Way. That book comes from Ireland which is located in Europe and where there is an ethnic group of white people.
Yu et al. (2020)		(“*A Long Long Way*”, “country”, “*Ireland*”) (“*A Long Long Way*”, “followedBy”, “*The Secret Scripture*”)
Our Model	*HER+URLs*	(“*A Long Long Way*”, “country”, “*Ireland*”) (“*A Long Long Way*”, “followedBy”, “*The Secret Scripture*”) (“*The Secret Scripture*”, “publisher”, “*Faber and Faber*”)
	TER+URLs	(“*A Long Long Way*”, “country”, “*Ireland*”) (“*A Long Long Way*”, “followedBy”, “*The Secret Scripture*”)
Ground Truth		(“*A Long Long Way*”, “country”, “*Ireland*”) (“*A Long Long Way*”, “followedBy”, “*The Secret Scripture*”) (“*The Secret Scripture*”, “publisher”, “*Faber and Faber*”) (“*Ireland*”, “location”, “*Europe*”)
Sample 5		3Arena is located in Dublin, the Republic of Ireland, where Críona Ní Dhálaigh was Lord Mayor. The owner of 3Arena is Live Nation Entertainment.
Yu et al. (2020)		(“*Dublin*”, “country”, “*Republic of Ireland*”) (“*Dublin*”, “leaderName”, “*Críona Ní Dhálaigh*”) (“*Dublin*”, “leaderName”, “*Lord Mayor*”)
Our model	*HER+URLs*	(“*3Arena*”, “location”, “*Dublin*”) (“*3Arena*”, “owner”, “*Live Nation Entertainment*”) (“*Dublin*”, “country”, “*Republic of Ireland*”)
	TER+URLs	(“*Dublin*”, “country”, “*Republic of Ireland*”) (“*Dublin*”, “leaderName”, “*Críona Ní Dhálaigh*”)
Ground Truth		(“*3Arena*”, “location”, “*Dublin*”) (“*3Arena*”, “owner”, “*Live Nation Entertainment*”) (“*Dublin*”, “country”, “*Republic of Ireland*”) (“*Dublin*”, “leaderName”, “*Críona Ní Dhálaigh*”)

Table 5.6: Prediction outputs on a few samples from the WebNLG test set.

First, in Sample 3, the *HE Extractor* in the model of Yu et al. (2020) missed the HE "Athens International Airport", thereby overlooking the valid triplet: ("*Athens International Airport*", "cityServed", "*Athens*") in the ground truth. Meanwhile, in our model, the entity "*Athens International Airport*" was detected by the *Entity Extractor*. Then, the *TER+URLs* submodule of the *HTER Extractor* identified this entity in the *head* role and extracted the triplet ("*Athens International Airport*", "cityServed", "*Athens*"). Additionally, we compared the outputs of the *HER+URLs* and *TER+URLs* submodules in our model. Although two triplets, namely, ("*Athens*", "country", "*Greece*") and ("*Greece*", "leaderName", "*Alexis Tsipras*"), were easily obtained by the two submodules, the *HER+URLs* submodule failed to extract the valid triplet: ("*Athens International Airport*", "cityServed", "*Athens*") when considering the entity "Athens" in the *tail* role. Thus, in this example, the *TER+URLs* submodule achieved a better result than that of the *HER+URLs* submodule.

Second, in Sample 4, the approach of Yu et al. (2020) omitted two valid triplets in the ground truth: ("*The Secret Scripture*", "publisher", "*Faber and Faber*") and ("*Ireland*", "location", "*Europe*"), because the *HE Extractor* missed two HEs: "The Secret Scripture" and "Ireland". In our model, although the module *Entity Extractor* could detect the entity "The Secret Scripture", its *TER+URLs* submodule failed to extract the triplet ("*The Secret Scripture*", "publisher", "*Faber and Faber*") when considering "The Secret Scripture" in the *head* role. Meanwhile, thanks to the *HER+URLs* submodule, it extracted this missed triplet by considering "Faber and Faber" in the *tail* role and detecting the corresponding HE "The Secret Scripture" with the relation type "publisher". Based on the outputs of the two submodules, it is clear that the *HER+URLs* submodule yielded a better result for this sample than that of the *TER+URLs* submodule.

Finally, in Sample 5, the model of Yu et al. (2020) obtained only two valid triplets in the ground truth: ("*Dublin*", "country", "*Republic of Ireland*") and ("*Dublin*", "leaderName", "*Críona Ní Dhálaigh*"). The *HE Extractor* of their model missed the HE "3Arena", thereby overlooking the triplets: ("*3Arena*", "location", "*Dublin*") and ("*3Arena*", "owner", "*Live Nation Entertainment*"). In our model, the *Entity Extractor* also missed the entity: "3Arena". Consequently, the *TER+URLs* submodule also overlooked the triplets: ("*3Arena*", "location", "*Dublin*") and ("*3Arena*", "owner", "*Live Nation Entertainment*"). Meanwhile, the *HER+URLs* submodule identified the entity "Dublin" in the *tail* role to extract the triplet ("*3Arena*", "location", "*Dublin*") and also identified the entity "Live Nation Entertainment" in the *tail* role to extract the triplet ("*3Arena*", "owner", "*Live Nation Entertainment*"). However, the *HER+URLs* submodule missed the valid triplet ("*Dublin*", "leaderName", "*Críona Ní Dhálaigh*"), whereas this triplet was detected by the *TER+URLs* submodule. Our model obtained the final result by combining the outputs of the two submodules.

We further consider the system performance of the predicted outputs of the (*HER+URLs* and *TER+URLs*) submodules of the *HTER Extractor* of our model on the entire WebNLG test set in Table 5.7. We can see that the number of predicted triplets by the *HER+URLs* submodule is $1,510$, whereas this number is 1530 for the *TER+URLs* submodule. In addition, these two submodules share $1,368$ common predicted triplets. Thus, the overlap percentage of the output of the *HER+URLs* submodule is 90.6, whereas this rate is 89.4 for the output of the *TER+URLs* submodule. In Table 5.7, our model achieved the best performance when combining the predicted outputs of the two submodules.

	F1
HER+URLs	85.1
TER+URLs	85.3
Combined	**86.0**

Table 5.7: Results of the performance analysis of the two submodules of our model on the WebNLG test set. The set *URLs* was created using Algorithm 1.

On the basis of the results of the analysis of the examples in Table 5.6 and of the performances of the submodules of our model in Table 5.7, we conclude that the "*exhaustive search*" strategy of our model is effective in solving the entity and relation extraction task.

5.3.4 Impact of Using Pre-trained Language Models

For a fair comparison, like in Yu et al. (2020), we did not exploit the advantages of using pretrained language models. In reality, a well-known pretrained language model named BERT was first proposed by Devlin et al. (2019). It has been widely applied to various NLP downstream tasks and has achieved considerable success. For the entity and relation extraction task, Hang et al. (2021) presented a BERT-based model named BERT-JEORE and obtained superior performance. Therefore, we further investigated the impact of using pretrained language models when they were used in our model.

Specifically, for our model in Figure 5.2, we replaced only the first BiLSTM encoder with a pretrained BERT-Base encoder to extract the representations of the original words from the input sample. Note that the BERT model first uses its tokenizer to split each original word into tokens (if necessary) and then outputs the vectors of these tokens. Thus, we obtained the representation of each original word by averaging its start token vector and its end token vector. In Table 5.8, we report the system performance of our model on the NYT and WebNLG datasets.

Model	**NYT**			**WebNLG**		
	Precision	*Recall*	*F1*	*Precision*	*Recall*	*F1*
BERT-JEORE (Hang et al., 2021)	88.5	84.6	86.5	79.1	91.4	84.8
Ours$_{LSTM}$	82.2	88.2	85.1	84.3	87.7	86.0
Ours$_{BERT}$	90.7	92.6	**91.6**	85.0	87.9	**86.4**

Table 5.8: Impact of using a pretrained BERT encoder in our model.

Clearly, our model showed further performance boost on both the NYT and WebNLG datasets when employing a pretrained BERT encoder, significantly improving the $F1$-score by 6.5 points on the NYT dataset. Compared to a recent model based on BERT (BERT-JEORE) by Hang et al. (2021) for the entity and relation extraction task, our BERT-based model achieved a better performance on the NYT and WebNLG datasets, with gains of 5.1 points and 1.6 points in the $F1$-score, respectively. In addition, it is interesting to note that, even without using the BERT encoder, our model still outperformed the recent model on the WebNLG test set. This indicates that our approach with a new decomposition strategy is simple but very effective in solving the entity and relation extraction task.

5.4 Conclusion and Future Work

This chapter proposes a new decomposition strategy along with a corresponding model framework for the joint entity and relation extraction task. Our approach mainly focuses on solving the overlapping triplet problem, one of the biggest challenges of this task, as only a few existing works can tackle this problem effectively. Our model uses a module to extract all the relevant entities, and for each extracted entity, another module is utilized to consider its *head/tail* entity roles and extract all the related triplets. In addition, the use of *URLs* helps to sufficiently deal with the sparse label problem of relation types in the same entity pair (e.g., EPO cases), which can be prevalent in this task. Experimental results on the two widely used datasets (NYT and WebNLG) showed that our model achieved a notable performance compared with a recent work (Hang et al., 2021). The results of further analysis experiments showed the effectiveness of our approach in handling overlapping and multiple triplet extraction scenarios.

Our proposed methodology has considerable potential for practical NLP applications such as information extraction, knowledge base population, and question answering. Moreover, the idea of using *URLs* may be relevant and promising for multilabel classification problems in general, not just for a specific task such as the entity and relation extraction task. In future work, we plan to apply this idea to the text classification task. Additionally, we also would like to introduce other methods for solving the overlapping triplet problem more effectively, such as considering how to change the weight of a label depending on whether it is a subset of the true label, and integrating available knowledge bases of entities into current models for boosting the system performance.

Chapter 6

Conclusion

6.1 Summary of Research Results

For the overview of my research, we focused on relation extraction task and investigated the task in three supervised approaches: "*fully-supervised relation extraction*", "*zero-shot relation extraction*", and "*end-to-end relation extraction*". While my Master's thesis concentrated on the perspective of "*fully-supervised relation extraction*", this dissertation devoted on the two remaining perspectives. Specifically, we proposed several methods to improve performance on "*zero-shot relation extraction*" in Chapters 3 and 4, while Chapter 5 devoted to "*end-to-end relation extraction*".

In Chapter 3, we presented a method improving discriminative learning for zero-shot relation extraction. This aspect is overlooked in previous works. Thus, we investigated if discriminative learning can help improve task performance. Our method incorporated discriminative embedding learning for both sentences and semantic relations. It guaranteed two important properties of embedding representations: *intra-relation compactness* and *inter-relation separability*, thereby enhancing the quality of sentence and relation embeddings. Experimental results on two benchmark datasets showed that the proposed method significantly outperforms the state-of-the-art methods. Additionally, visualizing the testing sentence embeddings produced by the state-of-the-art model and our model in Figure 3.3 indicated the better quality of the sentence embeddings generated by our model.

In Chapter 4, we proposed a new method to improve performance on zero-shot relation extraction. We argued that enhancing the semantic correlation between instances and relations is the key to drastically improving the performance of ZSRE. A new model entirely devoted to this goal through three main aspects was proposed: learning high-quality relation representation, designing strategic mini-batches, and binding two-way semantic distribution consistency. Specifically, our model acquired meaningful and high-quality representations for instances and relations in the first aspect. This aspect plays an essential role in understanding the semantic correlation between instances and relations. Second, we designed each mini-batch as a mini-task, including K different seen relations and K corresponding instances (K is a hyperparameter), and forced the model to pair them exactly. This strategy encourages the model to grasp the semantic relationship between instances and relations deeply. Finally, to fully exploit the semantic relationship between instances and relations, we use two-way interaction, which grasps the interaction not only "from each instance to relations" but also "from each relation to instances" and constrains the consistency of the two interaction distributions. Extensive experiments on two benchmark datasets have demonstrated the effectiveness and robustness of our proposed model, particularly in limited training data scenarios.

In Chapter 5, we concentrated on "*end-to-end relation extraction*", which aims to jointly extract entities and their semantic relations in text. We introduced a new decomposition strategy along with a corresponding model framework for this joint entity and relation extraction task. Our approach mainly focused on solving the overlapping triplet problem, one of the biggest challenges of this task, as only a few existing works can tackle this problem effectively. Our model used a module to extract all the relevant entities, and for each extracted entity, another module is utilized to consider its *head/tail* entity roles and extract all the related triplets. In addition, the use of "*unified relation labels*" set helped to sufficiently deal with the sparse label problem of relation types in the same entity pair (e.g., EPO cases), which can be prevalent in this task. Experimental results on the two widely used datasets (NYT and WebNLG) showed that our model achieved a notable performance compared with the state-of-the-art model. The results of further analysis experiments demonstrated the effectiveness of our approach in handling overlapping and multiple triplet extraction scenarios.

6.2 Open Problems and Future Work

In this dissertation, we made efforts to solve relation extraction in two supervised approaches: "*zero-shot relation extraction*" and "*end-to-end relation extraction*". Although the performance task on the two approaches is significantly improved, it still has some remaning problems, and we plan to resolve them in our future work. Specifically, they are two main issues as follows:

First, for "*zero-shot relation extraction*", we assumed that the set of *seen* relation labels ($\mathcal{Y}_\mathcal{S}$) in training stage and the set of *unseen* relation labels ($\mathcal{Y}_\mathcal{U}$) in testing stage are disjoint, *i.e.*, $\mathcal{Y}_\mathcal{S} \cap \mathcal{Y}_\mathcal{U} = \emptyset$. Here, we consider training phase to testing phase as: $\mathcal{Y}_\mathcal{S} \rightarrow \mathcal{Y}_\mathcal{U}$. Following this setting, a testing sentence will be classified in one of unseen relations of $\mathcal{Y}_\mathcal{U}$. However, it is more generalized and realistic when assuming a testing sentence may express semantic relation which can belong to $\mathcal{Y}_\mathcal{U}$ or $\mathcal{Y}_\mathcal{S}$, thereby setting from the training phase to the testing phase as: $\mathcal{Y}_\mathcal{S} \rightarrow \mathcal{Y}_\mathcal{S} \cup \mathcal{Y}_\mathcal{U}$. Following this new setting, the task is called "*generalized zero-shot relation extraction*" (GZSRE), where a model is trained on labeled sentences of the seen relations but then targeted to predict both seen and unseen relations for testing sentences. Intuitively, the new task GZSRE is more challenging but relevant for real-world scenarios. For the preliminary measures, we use our proposed models for ZSRE to tackle GZSRE and further propose more effective models in future work.

Second, for both the supervised approaches: "*zero-shot relation extraction*" and "*end-to-end relation extraction*", we tested our proposed methods on benchmark datasets in the the general domain. Although our methods effectively improve the task performance significantly, they might not work well in some specialized domains like the biomedical domain. Thus, we plan to evaluate our methods in such domains in future work. Additionally, our study mainly focused on intra-sentence relation extraction, where entities with their relations appear in the same sentence. In fact, entities may hold semantic relation over sentences. This phenomenon has become more and more popular in real-world scenarios. For example, in the document: "*[John Stanistreet]$_{e1}$ was an Australian politician. He was born in [Bendigo]$_{e2}$ to legal manager John Jepson Stanistreet and Maud Mcllroy.*". The semantic relation between the first entity "*John Stanistreet*" and the second entity "*Bendigo*" is *place_of_birth*. Therefore, we plan to tackle inter-sentence relation extraction as part of our future work. Specifically, we first extend our current works to solve the document-level relation extraction task, which is more challenging but might be helpful for real-world scenarios. However, it has some limitations to our current sentence-level RE methods when adapting them to solve the document-level RE task. For example, in the sentence-level context, entities are often close to each other and express their semantic relations (if any) in a simple and explicit manner. Conversely, in the document-level context, two entities can be very far from each other, thereby challenging our models to profoundly grip semantic relations expressed in an implicit and complex manner. We will need to consider this limitation carefully in solving the document-level RE task in our future work.

Considering the two open problems above, we plan to investigate and solve these problems in our future work. The final target is to resolve the relation extraction task more effectively, thereby benefiting related NLP applications such as information extraction, knowledge base construction, and question answering.

Finally, in this study, we proposed an improved decomposition strategy for joint entity and relation extraction in Chapter 5. However, this method only works in a supervised learning manner requiring the given training dataset. In fact, such training datasets are not always available in real-world scenarios, especially in some specialized domains like the biomedical domain. Meanwhile, we expect to build systems that can automatically extract entities and relations jointly without requiring any training corpus in the COVID-19 field. Specifically, due to the COVID-19 outbreak, it is essential to grasp valuable knowledge from a large number of COVID-19-related papers for dealing with the pandemic effectively. However, there is still a lack of a system that has the ability to automatically detect both entities with various types and their diverse relations through papers, especially when COVID-19 papers are published rapidly. This motivates us to build the CovRelex system (Tran et al., 2021), which aims to exploit such information.

The overview of the CovRelex system is introduced in Figure 6.1. It consists of five main modules: **Relation Extraction**, **Entity Recognition**, **Relation Clustering**, **Relation Scoring**, and **Graph Construction**. Now, we briefly introduce each of them. For the **Relation Extraction** module, we employ sev-

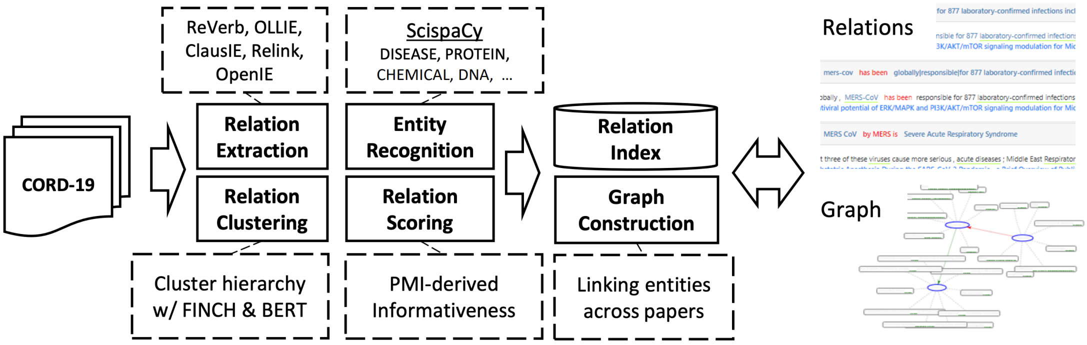

Figure 6.1: Overview of the CovRelex system.

eral relation extraction methods, including ReVerb (Fader et al., 2011), OLLIE (Mausam et al., 2012), ClausIE (Del Corro and Gemulla, 2013), OpenIE (Angeli et al., 2015), and ReLink (Tran and Nguyen, 2021). Meanwhile, for the **Entity Recognition** module, we use biomedical entity recognition models specialized for predicting entity type and provided by SciSpacy (Neumann et al., 2019). In the **Relation Clustering** module, we build a cluster hierarchy on a subset of the extracted relations using the clustering algorithm FINCH (Sarfraz et al., 2019), so users can quickly find their interesting relation expressions, or they can choose some clusters which may contain their interesting relation expressions. Besides, the **Relation Scoring** module is designed to calculate the informativeness of each relation, based on Pointwise Mutual Information (Church and Hanks, 1990). Finally, the **Graph Construction** module helps enable a more sophisticated paper search covering a complex graph describing relations among entities. The final goal of the CovRelex system is to automatically extract entities and their diverse relations not only in the same paper but also across many different papers.

Although the current CovRelex system helps support users in acquiring knowledge efficiently across a huge number of COVID-19 scientific papers published rapidly, it still has some challenges. First, the quality of relation extraction needs to be further improved. Second, the system should be able to solve the performance issue (e.g., the response time for user requests) when utilizing the present methods in the nick of time to fight pandemics. Therefore, we plan to improve the current CovRelex system according to the two challenges above, as the system is expected to be more effective and efficient for users in fighting with the coronavirus pandemic.

Publications

1. Van-Hien Tran, Van-Thuy Phi, Hiroyuki Shindo, and Yuji Matsumoto. 2019. Relation Classification Using Segment-Level Attention-based CNN and Dependency based RNN. 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 2793-2798, Minneapolis, Minnesota. Association for Computational Linguistics.

2. Van-Thuy Phi, Joan Santoso, Van-Hien Tran, Hiroyuki Shindo, Masashi Shimbo, and Yuji Matsumoto. 2019. Distant Supervision for Relation Extraction via Piecewise Attention and Bag-Level Contextual Inference. *IEEE Access* 7 (2019). DOI: 10.1109/ACCESS.2019.2932041

3. Vu Tran, Van-Hien Tran, Phuong Minh Nguyen, Phuong Chau Nguyen, Ken Satoh, Yuji Matsumoto, Minh Le Nguyen. 2020. CovRelex: A COVID-19 Retrieval System with Relation Extraction. The Fourth International Workshop on SCIentific DOCument Analysis, 2020.

4. Van-Hien Tran, Van-Thuy Phi, Akihiko Kato, Hiroyuki Shindo, Taro Watanabe, and Yuji Matsumoto. 2021. Improved Decomposition Strategy for Joint Entity and Relation Extraction. *Journal of Natural Language Processing*, 28(4):965-994.

5. Vu Tran, Van-Hien Tran, Phuong Nguyen, Chau Nguyen, Ken Satoh, Yuji Matsumoto, and Minh Nguyen. 2021. CovRelex: A COVID-19 Retrieval System with Relation Extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 24-31, Online. Association for Computational Linguistics.

6. Van-Hien Tran, Hiroki Ouchi, Taro Watanabe, and Yuji Matsumoto. 2022. Improving Discriminative Learning for Zero-Shot Relation Extraction. In *Proceedings of the 1st Workshop on Semiparametric Methods in NLP: Decoupling Logic from Knowledge, ACL 2022*, pages 1-6, Ireland and Online.

Association for Computational Linguistics.

7. Van-Hien Tran, Hiroki Ouchi, Hiroyuki Shindo, Yuji Matsumoto, and Taro Watanabe. 2022. Enhancing Semantic Correlation between Instances and Relations for Zero-Shot Relation Extraction. (Under Review, *Submitted to Journal of Natural Language Processing*)

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China. Association for Computational Linguistics.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019a. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019b. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications*, 114:34–45.

Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Yee Seng Chan and Dan Roth. 2010. Exploiting background knowledge for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 152–160, Beijing, China. Coling 2010 Organizing Committee.

Yee Seng Chan and Dan Roth. 2011. Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 551–560, Portland, Oregon, USA. Association for Computational Linguistics.

Chih-Yao Chen and Cheng-Te Li. 2021. ZS-BERT: Towards zero-shot relation extraction with attribute representation learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3470–3479, Online. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, Vancouver, Canada. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Yejin Choi, Eric Breck, and Claire Cardie. 2006. Joint extraction of entities and relations for opinion recognition. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 431–439.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Dai Dai, Xinyan Xiao, Yajuan Lyu, Shan Dou, Qiaoqiao She, and Haifeng Wang. 2019. Joint extraction of entities and overlapping relations using position-attentive sequence labeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6300–6308.

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: Clause-based open information extraction. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, page 355–366, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kathrin Eichler, Feiyu Xu, Hans Uszkoreit, Leonhard Hennig, and Sebastian Krause. 2016. TEG-REP: A corpus of textual entailment graphs based on relation extraction patterns. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3367–3372, Portorož, Slovenia. European Language Resources Association (ELRA).

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 2019. GraphRel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1409–1418, Florence, Italy. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

Jiaying Gong and Hoda Eldardiry. 2021. Prompt-based zero-shot relation classification with semantic knowledge augmentation. *CoRR*, abs/2112.04539.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention guided graph convolutional networks for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 241–251, Florence, Italy. Association for Computational Linguistics.

Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2537–2547, Osaka, Japan. The COLING 2016 Organizing Committee.

Jiale Han, Bo Cheng, and Guoshun Nan. 2021. Learning discriminative and unbiased representations for few-shot relation extraction. In *Proceedings of the 30th ACM International Conference on Information amp; Knowledge Management*, CIKM '21, page 638–648, New York, NY, USA. Association for Computing Machinery.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Tingting Hang, Jun Feng, Yirui Wu, Le Yan, and Yunfeng Wang. 2021. Joint extraction of entities and overlapping relations using source-target entity labeling. *Expert Systems with Applications*, 177:114853.

Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. 2017. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 221–231, Vancouver, Canada. Association for Computational Linguistics.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

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

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–122.

Jing Jiang and ChengXiang Zhai. 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, Rochester, New York. Association for Computational Linguistics.

Nanda Kambhatla. 2004a. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 22. Association for Computational Linguistics.

Nanda Kambhatla. 2004b. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 178–181, Barcelona, Spain. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Arzoo Katiyar and Claire Cardie. 2017. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 917–928, Vancouver, Canada. Association for Computational Linguistics.

Mahdy Khayyamian, Seyed Abolghasem Mirroshandel, and Hassan Abolhassani. 2009. Syntactic tree-based relation extraction using a generalization of Collins and Duffy convolution tree kernel. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Student Research Workshop and Doctoral Consortium*, pages 66–71, Boulder, Colorado. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Hoang-Quynh Le, Duy-Cat Can, Sinh T. Vu, Thanh Hai Dang, Mohammad Taher Pilehvar, and Nigel Collier. 2018. Large-scale exploration of neural relation classification architectures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2266–2277, Brussels, Belgium. Association for Computational Linguistics.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 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.

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

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2020. Distilling dense representations for ranking using tightly-coupled teachers. *arXiv preprint arXiv:2010.11386*.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. 2002. Text classification using string kernels. *Journal of machine learning research*, 2(Feb):419–444.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

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

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

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

Raymond Mooney and Razvan Bunescu. 2005. Subsequence kernels for relation extraction. *Advances in neural information processing systems*, 18.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and robust models for biomedical natural language processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy. Association for Computational Linguistics.

Dat P.T. Nguyen, Yutaka Matsuo, and Mitsuru Ishizuka. 2007. Subtree mining for relation extraction from Wikipedia. In *Human Language Technologies*

2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers, pages 125–128, Rochester, New York. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2014. Employing word representations and regularization for domain adaptation of relation extraction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 68–74, Baltimore, Maryland. Association for Computational Linguistics.

Thien Huu Nguyen, Barbara Plank, and Ralph Grishman. 2015. Semantic representations for domain adaptation: A case study on the tree kernel-based method for relation extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 635–644, Beijing, China. Association for Computational Linguistics.

Abiola Obamuyide and Andreas Vlachos. 2018. Zero-shot relation classification as textual entailment. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 72–78, Brussels, Belgium. Association for Computational Linguistics.

Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. 2018. A span selection model for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1630–1642, Brussels, Belgium. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Dejing Dou, and Thien Huu Nguyen. 2020. Exploiting the syntax-model consistency for neural relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8021–8032, Online. Association for Computational Linguistics.

Pengda Qin, Weiran Xu, and Jun Guo. 2016. An empirical convolutional neural network approach for semantic relation classification. *Neurocomputing*, 190:1–9.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2020. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2010.08191*.

Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11:2487–2531.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R. Voss, Heng Ji, Tarek F. Abdelzaher, and Jiawei Han. 2017. Cotype: Joint extraction of typed entities and relations with knowledge bases. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1015–1024, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163, Berlin, Heidelberg. Springer Berlin Heidelberg.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Benjamin Rosenfeld and Ronen Feldman. 2007. Using corpus statistics on entities to improve semi-supervised relation extraction from the web. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 600–607, Prague, Czech Republic. Association for Computational Linguistics.

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

M. Saquib Sarfraz, Vivek Sharma, and Rainer Stiefelhagen. 2019. Efficient parameter-free clustering using first neighbor relations. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8926–8935.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Yu-Ming Shang, Heyan Huang, and Xianling Mao. 2022. Onerel: Joint entity and relation extraction with one module in one step. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11285–11293.

Sameer Singh, Sebastian Riedel, Brian Martin, Jiaping Zheng, and Andrew McCallum. 2013. Joint inference of entities, relations, and coreference. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 1–6.

Daniil Sorokin and Iryna Gurevych. 2017. Context-aware representations for knowledge base relation extraction. In *Proceedings of the 2017 Conference on*

Empirical Methods in Natural Language Processing, pages 1784–1789, Copenhagen, Denmark. Association for Computational Linguistics.

Axel J Soto, Piotr Przybyła, and Sophia Ananiadou. 2019. Thalia: semantic search engine for biomedical abstracts. *Bioinformatics*, 35(10):1799–1801.

Ang Sun, Ralph Grishman, and Satoshi Sekine. 2011. Semi-supervised relation extraction with large-scale word clustering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 521–529, Portland, Oregon, USA. Association for Computational Linguistics.

Xu Sun, Wenjie Li, Houfeng Wang, and Qin Lu. 2014. Feature-frequency–adaptive on-line training for fast and accurate natural language processing. *Computational Linguistics*, 40(3):563–586.

Idan Szpektor, Hristo Tanev, Ido Dagan, and Bonaventura Coppola. 2004. Scaling web-based acquisition of entailment relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 41–48, Barcelona, Spain. Association for Computational Linguistics.

Yuanhe Tian, Guimin Chen, Yan Song, and Xiang Wan. 2021. Dependency-driven relation extraction with attentive graph convolutional networks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4458–4471, Online. Association for Computational Linguistics.

Van-Hien Tran. 2019. Relation classification using segment-level attention-based cnn and dependency rnn. Master's thesis, Nara Institute of Science and Technology, Ikoma, Japan.

Van-Hien Tran, Van-Thuy Phi, Hiroyuki Shindo, and Yuji Matsumoto. 2019. Relation classification using segment-level attention-based CNN and dependency-based RNN. 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 2793–2798, Minneapolis, Minnesota. Association for Computational Linguistics.

Vu Tran, Van-Hien Tran, Phuong Nguyen, Chau Nguyen, Ken Satoh, Yuji Matsumoto, and Minh Nguyen. 2021. CovRelex: A COVID-19 retrieval system with relation extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 24–31, Online. Association for Computational Linguistics.

Xuan-Chien Tran and Le-Minh Nguyen. 2021. Relink: Open information extraction by linking phrases and its applications. In *International Conference on Distributed Computing and Internet Technology*, pages 44–62. Springer.

Ngoc-Trinh Vu, Van-Hien Tran, Thi-Huyen-Trang Doan, Hoang-Quynh Le, and Mai-Vu Tran. 2015. A method for building a labeled named entity recognition corpus using ontologies. In *Advanced Computational Methods for Knowledge Engineering*, pages 141–149, Cham. Springer International Publishing.

Haoyu Wang, Ming Tan, Mo Yu, Shiyu Chang, Dakuo Wang, Kun Xu, Xiaoxiao Guo, and Saloni Potdar. 2019. Extracting multiple-relations in one-pass with pre-trained transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1371–1377, Florence, Italy. Association for Computational Linguistics.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721, Online. Association for Computational Linguistics.

Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. 2016. Relation classification via multi-level attention CNNs. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1307, Berlin, Germany. Association for Computational Linguistics.

Shusen Wang, Bosen Zhang, Yajing Xu, Yanan Wu, and Bo Xiao. 2022. RCL: Relation contrastive learning for zero-shot relation extraction. In *Findings* *of the Association for Computational Linguistics: NAACL 2022*, pages 2456–2468, Seattle, United States. Association for Computational Linguistics.

Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. 2013. Pubtator: a web-based text mining tool for assisting biocuration. *Nucleic acids research*, 41(W1):W518–W522.

Ji Wen. 2017. Structure regularized bidirectional recurrent convolutional neural network for relation classification. *arXiv preprint arXiv:1711.02509*.

Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. 2016. A discriminative feature learning approach for deep face recognition. In *Computer Vision – ECCV 2016*, pages 499–515. Springer International Publishing.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 2361–2364.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2015. Semantic relation classification via convolutional neural networks with simple negative sampling. *arXiv preprint arXiv:1506.07650*.

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question answering on Freebase via relation extraction and textual evidence. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2326–2336, Berlin, Germany. Association for Computational Linguistics.

Bishan Yang and Claire Cardie. 2013. Joint inference for fine-grained opinion extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1640–1649.

Yunlun Yang, Yunhai Tong, Shulei Ma, and Zhi-Hong Deng. 2016. A position encoding convolutional neural network based on dependency tree for relation classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 65–74, Austin, Texas. Association for Computational Linguistics.

Bowen Yu, Zhenyu Zhang, Xiaobo Shu, Tingwen Liu, Yubin Wang, Bin Wang, and Sujian Li. 2020. Joint extraction of entities and relations based on a novel decomposition strategy. *Frontiers in Artificial Intelligence and Applications*, 325(ECAI 2020):2282–2289.

Xiaofeng Yu and Wai Lam. 2010. Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach. In *Coling 2010: Posters*, pages 1399–1407, Beijing, China. Coling 2010 Organizing Committee.

Yue Yuan, Xiaofei Zhou, Shirui Pan, Qiannan Zhu, Zeliang Song, and Li Guo. 2020. A relation-specific attention network for joint entity and relation extraction. In *IJCAI*, volume 2020, pages 4054–4060.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *J. Mach. Learn. Res.*, 3(null):1083–1106.

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

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 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.

Wenyuan Zeng, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2017. Incorporating relation paths in neural relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1768–1777, Copenhagen, Denmark. Association for Computational Linguistics.

Xiangrong Zeng, Shizhu He, Daojian Zeng, Kang Liu, Shengping Liu, and Jun Zhao. 2019. Learning the extraction order of multiple relational facts in a sentence with reinforcement learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 367–377, Hong Kong, China. Association for Computational Linguistics.

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

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, M. Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

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

Min Zhang, Jie Zhang, and Jian Su. 2006a. Exploring syntactic features for relation extraction using a convolution tree kernel. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 288–295, New York City, USA. Association for Computational Linguistics.

Min Zhang, Jie Zhang, Jian Su, and Guodong Zhou. 2006b. 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.

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

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 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.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

Tianyang Zhao, Zhao Yan, Yunbo Cao, and Zhoujun Li. 2020. Asking effective and diverse questions: a machine reading comprehension based framework for joint entity-relation extraction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 3948–3954.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1227–1236, Vancouver, Canada. Association for Computational Linguistics.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 50–61, Online. Association for Computational Linguistics.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 427–434, Ann Arbor, Michigan. Association for Computational Linguistics.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Berlin, Germany. Association for Computational Linguistics.

Wenxuan Zhou and Muhao Chen. 2021. An improved baseline for sentence-level relation extraction. *arXiv preprint arXiv:2102.01373*.