UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Ontological Knowledge Learning for Relation Extraction

by Yang Li

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of A/Prof. Guodong Long
A/Prof. Jing Jiang
Dr. Angela Huo

University of Technology Sydney
Faculty of Engineering and Information Technology

July 2023

Certificate of Original Authorship

I, *Yang Li*, declare that this thesis is submitted in fullfilment of the requirements for the award of *Doctor of Philosophy*, in the Australian Artificial Intelligence Institute, Faculty of Engineering and Information Technology, at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note: Signature removed prior to publication.

Date: January 23, 2024

ABSTRACT

by

With the popularity of the internet, an unprecedented amount of digital text has been generated every day in various forms. This unstructured or semi-structured text contains a huge amount of information. To further employ such information, Information Extraction (IE) has attracted more and more attention. IE can extract meaningful information from plain text and stores it in a structured format. Relation Extraction (RE), as one of the most important sub-tasks of IE, can identify relationships between given entities. Many Natural Language Processing (NLP) tasks can benefit from this extracted relational information, including search engines, Knowledge Graphs (KGs), Information Retrieval, Query Understanding, Question-Answering Systems, etc.

RE aims to discriminate the relation between two given entities in plain text. With the development of Deep Learning, data-driven algorithms, e.g., Deep Neural Networks (DNNs), have become the major approaches for Artificial Intelligence (AI) Tasks. Although DNNs have strong capabilities of comprehending sentence semantics in tackling many NLP tasks, it is hard to grasp the ontological knowledge of named entities. In order to train a reliable model for the RE task, a substantial volume of training data is required. Nevertheless, obtaining the data through crowd-sourcing annotations is a laborious and time-consuming process.

Thus, the Distant Supervision (DS) method is proposed to automatically annotate training data by aligning named entities in existing KGs based on a strong assumption. This strong assumption has inevitably caused the wrong labeling problem. Besides, KGs usually suffer from long-tail relations due to its incompleteness. The long-tail relations in RE lead to fewer training sentences, which seriously disrupts the data balance. Thus, the wrong-labeling problem and the long-tail relations are the two main challenges to further take a step in Distant Supervision Relation Extraction (DSRE).

For these two main challenges, existing works specifically embed the relative distance embedding of two named entities to learn ontological knowledge. Then, the Multi-Instance Learning (MIL) framework is proposed to relieve the strong assumption. Combined with the selective attention networks, the MIL framework can further exploit valid information from noisy sentences. Some works further introduce extra ontological knowledge from KGs to enrich entity pairs or leverage the relational hierarchy.

Former works highly rely on the selective attention network to denoise the wrongly labeled sentences in each bag. However, these works cannot tackle the bag with only one sentence, or even one sentence with the wrong label. We propose a brand-new light-weight framework to further exploit the ontological knowledge in the plain text.

For the long-tail relations, former works naturally leverage the relational hierarchy to share the knowledge from data-rich relations to the long-tail one when those relations have semantic overlap or introduce extra KGs. We propose collaborating relation-augmented attention to infuse relational knowledge by cross-relation sharing. To further mitigate two mentioned challenges, we leverage extra ontological knowledge in KGs and align multi-granular entity types with sentences.

Besides, we also target to solve the RE task with insufficient truly labeled training data. Meanwhile, the Pre-trained Language Models (PLMs) become the master of various NLP tasks. These PLMs are trained by transfer learning with massive plain text. Nonetheless, to transfer the learning knowledge from PLMs to RE tasks, the fine-tuning paradigm is indispensable and needs a large scale of truly labeled training data. Thus, we employ the prompt tuning to overcome the lack of data. We further imitate the human decision process to exploit the ontological knowledge from relations which aims to find the contrastive attributes between the object factual and their potential counterfactuals.

Dissertation directed by A/Prof. Guodong Long, A/Prof. Jing Jiang and Dr. Angela Huo
Australian Artificial Intelligence Institute
Faculty of Engineering and IT
University of Technology Sydney

Acknowledgements

First of all, I am extremely grateful to my principal supervisor A/Prof. Guodong Long, and two co-supervisors A/Prof. Jing Jiang and Dr Huan Huo. Their immense knowledge and plentiful experience have encouraged me in all the time of my academic research and daily life. During the stage of chasing my Ph.D. degree, my principal supervisor A/Prof. Guodong Long is the guide on my academic journey and generously imparts a lot of experience in academics to me. A/Prof. Jing Jiang not only focuses on my research progress but also cares about my life in Sydney. On holidays and leisure time, she organizes team activities to enhance team cohesion and promote our physical and mental health. Their mentorship has truly enriched my educational experience. Additionally, I wish to express my profound gratitude to Dr. Huan Huo and Prof. Chengqi Zhang for providing me with the opportunity to study at the University of Technology Sydney.

I also could not have undertaken this journey evenly without my senior mate Dr. Tao Shen, who generally shares knowledge and expertise with me. I am also thankful to my teammates: Hao Huang, Zonghan Wu, Fengwen Chen, Lu Liu, Haiyan Zhao, Zhuowei Wang, Shuang Ao, Yue Tan, Jie Ma, Peng Yan and Dr. Xueping Peng. I am also grateful to my mentors during my internship, Canran Xu and Fangfang Li.

Lastly, I would like to mention my parents, Li Yang and Zhengxiang Li. I would not achieve such progress without their support, education, and solicitude. Specifically, thanks should also go to my wife, Weiyi Lu, who companies with me day and night on this journey and bear with my mood swings. Last of last, thanks to every teaches who has taught me; thanks to all my friends who share happiness and pair, and companies with me; thanks to all the lovely people who have helped me and affected me positively.

Yang Li

Sydney, Australia, 2023

List of Publications

Conference Papers

C-1. **Yang Li**, Guodong Long, Tao Shen, Tianyi Zhou, Lina Yao, Huan Huo, and Jing Jiang: Self-attention enhanced selective gate with entity-aware embedding for distantly supervised relation extraction. The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020: 8269–8276.

Core Rank: **A***

C-2. **Yang Li**, Tao Shen, Guodong Long, Jing Jiang, Tianyi Zhou, and Chengqi Zhang: Improving Long-Tail Relation Extraction with Collaborating Relation-Augmented Attention. Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020: 1653–1664.

Core Rank: **A**

C-3. **Yang Li**, Guodong Long, Tao Shen, and Jing Jiang: Hierarchical Relation-Guided Type-Sentence Alignment for Long-Tail Relation Extraction with Distant Supervision. Findings of the Association for Computational Linguistics: NAACL 2022: 316–326.

Core Rank: **A**

C-4. **Yang Li**, Canran Xu, Tao, Shen, Jing Jiang, and Guodong Long. "CCPrefix: Counterfactual Contrastive Prefix-Tuning for Many-Class Classification," The 18th Conference of the European Chapter of the Association of Computational Linguistics, EACL 2024

Core Rank: **A**

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Nowadays, a huge amount of digital text has been recorded in various forms, e.g., reports, blogs, emails, papers, etc. The implicit knowledge maintained in this unstructured or semi-structured text is important because the extracted structured knowledge can be reused to improve the efficiency and performance of many downstream tasks.

Extracting structured knowledge from plain text is called Information Extraction (IE) [12, 13, 14]. The structured knowledge is diverse, e.g., named entities, relations, types, events, sentiments, etc. In this structured knowledge, named entities and relations are the most reusable information and can benefit various downstream Natural Language Processing (NLP) tasks, e.g., Knowledge Graphs (KGs), query understanding, information retrieval, question-answering systems, etc. A Named Entity (NE) is often a word or phrase representing a specific real-world object [15, 16, 17, 18, 19]. For example, Steve Jobs is an NE. In the following sentence "Steve Jobs founded Apple.", Steve Jobs is mentioned. In another sentence "Steve Jobs ate an apple today. He felt great.", i.e., Steve Jobs is totally mentioned twice. An NE mentioned in plain text can be shown in different formats, i.e., the name itself, nominal, or pronominal. Besides, NEs are often categorized into various generic types, e.g., `PERSON`, `LOCATION`, `COMPANY`, `DATE`, etc. It is worth noting that the same NE in different sentences may have completely unrelated generic types. In the former two examples, Apple appeared twice. Apple is `COMPANY` in the first sentence while it is `FRUIT` in the second sentence.

A relation usually represents a well-defined relationship between two given entities.

For example, /BUSINESS/COMPANY/FOUNDERS is the relation between PERSON and COMPANY. There is considerable interest in Relation Extraction (RE), both as an end in itself and as an intermediate step in a variety of natural language processing tasks.

The objective of the RE task is to accurately identify the relationship between two provided named entities with plain text sentences. Such an extracted relationship between two named entities can benefit the down-steam NLP tasks via integrating ontological knowledge. Specifically, an efficient RE model can identify that a person is employed by a particular organization, or that a geographic entity is located in a particular region [1, 20, 21]. For example, the raw sentence "Steve Jobs founded Apple" expresses that Steve Jobs is the founder of the company Apple. Thus, the relation between Steve Jobs and Apple can be represented as /BUSINESS/COMPANY/FOUNDERS.

In the last decade, deep learning has developed rapidly and achieved great success in a wide range of fields. The exponential growth in computing power and the unprecedented scale of data have been instrumental in driving the remarkable achievements of deep learning. The notable proliferation of computing power, accompanied by the exponential expansion of the scale of data, constitutes the pivotal driver of deep learning's exceptional performance. For various NLP tasks, DNNs have become the priority choice to learn task-specific distributed representations in an end-to-end fashion with State-Of-The-Art (SOTA) performance [22, 23]. In the context of NLP tasks, a deep learning algorithm is utilized through the following sequential steps. Firstly, the raw text is segmented into discrete tokens. Subsequently, each token is transformed into a dense, low-dimensional vector representation. These vectors are initially using pre-trained word embeddings such as GloVe [24] or word2vec [25]. Next, the distributed representation, comprising these token vectors, is fed into deep neural networks (DNNs). This process aims to generate a more robust sentence representation that can be utilized for various tasks.

Based on the quality of the dataset, there are three main training paradigms employed

in the field:

1. **Supervised paradigm** Supervised approaches [26, 27, 28] rely on annotated training data. However, human-annotated training data is expensive and thus limited in quantity. Moreover, supervised classifiers often exhibit a bias toward the specific domain they are trained on.

2. **Semi-supervised paradigm** Many semi-supervised approaches [29, 30] heavily rely on the bootstrap learning. They initially use a small dataset to learn how to extract additional relations, and then iteratively use the extracted relations for further training. Although semi-supervised approaches significantly reduce the need for manual efforts in creating training data, they still require an initial set of labeled pairs for relation.

3. **Unsupervised paradigm** Unsupervised approaches [31, 32, 33] do not require any labelled training data. However, they typically yield sub-optimal results that are challenging to interpret and map to existing relations, schemas, or ontologies. This limitation is particularly critical for applications like knowledge base refinement.

It is worth noting that the model performance is quite sensitive to not only the quantity but also the reliability of the annotated training data. For some specific domains, e.g., medical, mechanical and aerospace, the annotation of the dataset may require expert knowledge. In RE task, the label of each training sample is the relationship of two certain named entities. For human annotators, the named entities should be distinguished first. Then, the human annotators need to recognize the valid relationship between two named entities. To comprehend the context, two aspects of text should be considered by human annotators, i.e., syntax and semantics. Concretely, syntax stands for the arrangement of words in a sentence such that they make grammatical sense. While, semantics refers to the meaning contained in the plain text.

Table 1.1 : Sentences with the same two named entities, which two of them are correctly labeled and the other is wrongly labeled by distant supervision.

Sentences	**Label**	**Correct**
Barack Obama was the president of the United States.	PRESIDENT_OF	True
Barack Obama lives in the United States with his wift.	PRESIDENT_OF	False
Barack Obama visit China as the president of United States	PRESIDENT_OF	True

To address the scarcity of large annotated training data, the Distant Supervision (DS) method [34] has been introduced. Distant Supervision Relation Extraction (DSRE) leverages KGs to automatically annotate data, thereby combining the benefits of both semi-supervised and unsupervised RE approaches. KG transfers triple sets into a directed graph with nodes and edges. Nodes represent the named entities in the triples and edges represent the relationship between two nodes. For each named entity, KG always stores corresponding attribute information, e.g., type, occupation, and other fine-grained features. Distant supervision assumes that two named entities in various sentences may express the same relationship in existing KG. It is evident that the relationship between the same pair of named entities in different sentences may not consistently align with the relationships described in KG. Consequently, the strong assumption made by DS inherently leads to the problem of wrong labeling, as illustrated in Table 1.1. This issue arises due to the reliance on KGs to automatically assign labels, which may not always accurately reflect the relationships expressed in the text. Furthermore, due to the inherent completeness of KGs, the training samples automatically annotated using these KG may suffer from the issue of

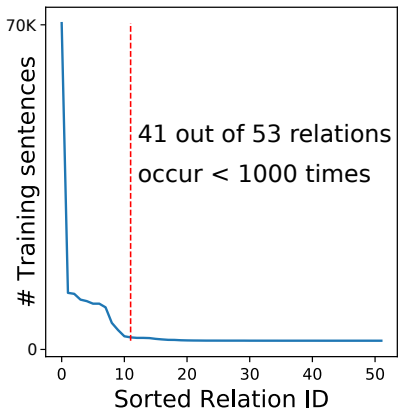

Figure 1.1 : Label frequency distribution of relations without *NA* in NYT dataset. Here the criterion being a long-tail relation is the number of corresponding training sentences is less than 1000.

long-tail relations. In other words, there are certain relations in the triple set of KGs that appear much less frequently in the corpus, resulting in a significantly lower number of training samples annotated with other specific relations compared to others. To illustrate this challenge, consider the construction of the New York Times (NYT) dataset [35] using Freebase as the annotator. As depicted in Figure 1.1, approximately 70% of the relations in the dataset fall under the long-tail category. This data imbalance issue severely hampers improvements in DSRE, posing a significant challenge to overcome.

As previously mentioned, DSRE encounters two significant challenges: 1) the wrong labeling problem; 2) the presence of long-tail relations. Multi-instance learning (MIL) framework [35, 36] is proposed to relax the strong assumption of DS to *at-least-one* assumption. In plainer terms, this means that any possible relation between two named entities is considered true in at least one distantly-labeled sentence, rather than requiring all sentences with the entity pair to express the relation. In this framework, instead of assigning sentence-level labels to individual samples, a label is assigned to a bag of sentences containing a common entity pair. The label represents the relationship between the

entity pair in the KGs. Some researchers [1, 21] utilize this framework to design task-specific modules and leverage the powerful capabilities of DNNs in capturing implicit semantics in the context. To mitigate the impact of wrongly labeled sentences, the selective attention [37] framework has been proposed. This framework selectively filters out relevant information from each bag of sentences to predict the relationship between two entities, heuristically reducing the influence of wrong labels. To tackle the long-tail relations, several approaches [4, 5] have exploited hierarchical structures to leverage the knowledge learned from data-rich relations and transfer it to long-tail relations with semantic overlap. This semantic overlap or relatedness is typically encoded in relation hierarchies present in KGs, such as Freebase [38]. These hierarchical structures enable the sharing of knowledge and information among different instances via their same superior relations, improving the extraction performance for long-tail relations.

Recently, some works [8,39,40,41,42,43] form the self-attention networks as the main encoder in Pre-trained Language Models (PLMs). These models undergo self-supervised pre-training on large-scale unlabeled corpora, such as Wikipedia, Book Corpus, and Common Crawl, to improve their understanding of the text and enhance their expressive power. The pre-training phase, performed on publicly available and easily accessible unlabeled corpora, significantly boosts the language comprehension capabilities of these models. Following pre-training, the fine-tuning paradigm is employed, where the PLMs transfer the learned knowledge to specific tasks using labeled task-specific data. This approach has achieved state-of-the-art performance on a wide range of NLP tasks [8, 39, 44, 45]. However, the performance of PLMs with the fine-tuning paradigm heavily relies on the quality of labeled task-specific training data. Moreover, the significant gap between the objective forms in pre-training and fine-tuning restricts the full utilization of the knowledge stored within PLMs. To address this issue, prompt-tuning has been proposed to bridge the gap between the objective forms in pre-training and fine-tuning [46]. In prompt-tuning, a set of label words and prompt templates are designed to create a cloze-style task that aligns

with the pre-training objective. For example, in natural language inference, the classes are entailment, neutrals, and contradiction. Some works [47,48] use {“yes”, “maybe”, “no”} as the set of label words. In binary sentiment classification, the positive sentiment and the negative sentiment are respectively mapped to “good” and “bad”. By incorporating these label words and prompt templates, prompt-tuning aims to align the objective forms between pre-training and fine-tuning, enhancing the performance of PLMs on specific tasks, such as classification, relation extraction, and so on.

In the context of the RE task, the primary focus of pre-training tasks in PLMs is to learn the semantic understanding of individual sentences. As a result, transferring semantic information into ontological knowledge using the fine-tuning approach becomes challenging. Additionally, even with the adoption of the prompt-tuning paradigm, there are still difficulties in compelling PLMs to effectively grasp and incorporate ontological knowledge in sparse data scenarios. Addressing this challenge requires exploring alternative approaches that can effectively leverage ontological knowledge, especially in scenarios with sparse data. Thus, further efforts are necessary to enhance the ability of PLMs to understand and extract ontological knowledge via prompt-tuning approach [46,47,49,50,51,52,53,54]. Brown et al. [46] assert that scaling up language models significantly enhances their ability to perform task-agnostic, few-shot learning, sometimes rivaling the effectiveness of previous state-of-the-art fine-tuning methods. Schick et al. [47] introduces an automated mapping technique that identifies correspondences between words and labels with minimal training data, reducing the reliance on domain expertise. Schick et al. [49] demonstrate that language models with significantly fewer parameters can achieve competitive performance by transforming textual inputs into cloze-style questions with task descriptions and employing gradient-based optimization, further enhanced by utilizing unlabeled data. Shin et al. [50] propose auto prompt for fine-tuning without extra parameters. Gao et al. [51] provides prompt-based fine-tuning with an automated prompt generation pipeline and a refined strategy for dynamically incorporating

demonstrations into contexts. Zhong et al. [52] propose opti-prompt, which optimizes within a continuous embedding space. Lester et al. [53] propose soft prompts to adapt fixed language models for specific downstream tasks. Li et al. [54] propose counterfactual contrastive prefix-tuning for many-class classification.

1.2 Research Problems

The field of Relation Extraction has encountered several challenges, particularly regarding the availability of reliable training data. Training a robust deep neural RE model requires a substantial amount of accurately annotated data, which is traditionally obtained through manual annotation. However, this process is extremely time-consuming and labor-intensive. Although the distant supervision method has been proposed as an alternative to automatically annotating training data, it introduces challenges such as wrong labeling and the prevalence of long-tail relations. Obviously, the development of RE is constrained by the quality and quantity of annotated training data.

In light of this, our research focuses on addressing the RE task under distant supervision, as well as the RE task with a small amount of data but high annotation accuracy. We aim to tackle the following four research problems through our investigation:

- **Research Problem 1**: The severe problem of wrong labeling in training data under the strong assumption of distant supervision. Existing works have attempted to mitigate this issue by employing the Multi-instance Learning framework [35, 36] in conjunction with selective attention networks [37]. However, these methods are vulnerable when a bag consists of a single sentence, and even worse, when a single sentence expresses inconsistent relation information with the bag-level label. This scenario is not uncommon in the popular RE dataset, NYT.

- **Research Problem 2**: The significant impact of long-tail relations on model performance due to the incompleteness of KGs in the distant supervision method.

Long-tail relations suffer from insufficient training samples, posing a challenge in providing sufficient information for training under such circumstances. To address this, we leverage the relational hierarchy and employ attention mechanisms to share knowledge from the data-rich relation to long-tail ones, ensuring that even limited training information can be fully utilized.

- **Research Problem 3**: Semantically aligning long-tail relation samples with coarse-grained relations solely based on sentence semantics is difficult. Inaccurate transfer of information from the data-rich relation to long-tail relations can accumulate errors and impact the identification of other classes. To overcome this challenge, we investigate methods to improve the alignment between sentences and coarse-grained relations.

- **Research Problem 4**: In scenarios with a small amount of training data but high annotation quality, such as few-shot learning settings, there is a significant challenge in transferring semantic knowledge from Pre-trained Language Models (PLMs) to ontological knowledge for the Relation Extraction (RE) task. To address this, we explore the prompt-tuning approach, aiming to elicit specific knowledge from PLMs by constructing appropriate prompt templates.

1.3 Research Objectives and Methods

In order to address the research objectives and challenges mentioned in this section, we present a brief overview of the methods employed for each objective:

1. For the first research objective, we focus on producing entity-aware embeddings and rich-contextual representations to enhance downstream aggregation modules. Additionally, we replace the previous selective attention with a gate mechanism and pooling layer to address one-sentence bags. Experimental evaluations on the NYT dataset demonstrate significant progress.

2. To further improve the performance of long-tail relations, we propose a relation-augmented attention mechanism within the relational hierarchy. This approach leverages high-level relations to collaboratively enhance features, taking into account the hierarchical structure.

3. We incorporate extra information related to named entities in a sentence to address the challenge of inferring coarse-grained relations solely based on semantics. Based on the extra information, we introduce two novel modules designed specifically to tackle the major challenges in DSRE.

4. To overcome the challenges posed by the DS method and the limited availability of labeled training data, we shift the focus to training datasets with accurate labels, even if the amount of labeled data falls short of fully supervised learning requirements. In data-scarce scenarios, we employ prompt tuning as an alternative to the fine-tuning paradigm.

We aim to make significant contributions to the field of RE, improving the robustness, performance, and applicability of RE under distant supervision and in scenarios with limited labeled training data.

1.3.1 Thesis Organization

In this section, we present an overview of the thesis organization, outlining the structure and content of each chapter:

- *Chapter 2*: This chapter presents a detailed literature review of milestone works in the RE task and several key text representation approaches, including attention mechanism, transformer, and masked language model.

- *Chapter 3*: This chapter presents our novel approach for DSRE. The focus is on addressing the wrong labeling problem in DSRE and proposing a specific embedding methodology that combines ontological knowledge and semantics. Detailed

experiments are conducted to evaluate the effectiveness and performance of our approach.

- *Chapter 4*: This chapter introduces a novel hierarchical attention network to tackle the challenge of long-tail relations in DSRE. The sentence representations are associated with relational knowledge, and the resulting representations are shared through a hierarchical structure. The chapter provides in-depth explanations of the proposed approach and presents experimental results to validate its efficacy.

- *Chapter 5*: This chapter presents two novel modules that incorporate extra ontological knowledge from existing KGs to enhance the text representation for DSRE in different ways. Both modules leverage the additional ontological knowledge to strengthen the representation of textual data, leading to improved performance in the RE task. The chapter includes comprehensive experiments to demonstrate the effectiveness of this approach.

- *Chapter 6*: This chapter introduces the concept of prompt-tuning with counterfactuals for the RE task in data-scarce scenarios. The approach involves generating prompt templates that incorporate contrastive knowledge between relations, aiming to inject ontological knowledge into PLMs. The chapter provides detailed explanations of the prompt-tuning methodology and presents experimental results to support its feasibility and effectiveness.

- *Chapter 7*: This chapter offers a summary of the research conducted in this thesis and discusses potential avenues for future work and research directions.

By following this organizational structure, the thesis aims to provide a comprehensive understanding of the challenges in RE, presents novel approaches to address these challenges, and offers insights for future research in this field.

Chapter 2

Literature Review

In this chapter, we undertake an extensive exploration into the complex dynamics of ontological knowledge learning and its critical impact on overcoming challenges in RE.

Our review begins by acknowledging the foundational work in the field, highlighting the pivotal works on extracting ontological knowledge via Neural Networks (NNs) to enhance the precision and understanding of RE tasks, such as piecewise max-pooling convolutional neural network, hierarchical structure, and KGs-integrated architecture. We then continue our narrative by dissecting the evolution and impact of attention mechanisms, examining everything from Vanilla to Selective and Multi-level Attention Networks, alongside the Transformer. In this section, we underscore the advancements in attention-based models. Vanilla attention mechanisms provided a solid foundation, upon which various attention structures have been derived to meet the unique needs of RE tasks for ontological knowledge. For instance, by incorporating hierarchical structures, hierarchical attention has been proposed to better capture and utilize ontological knowledge. Similarly, the development of multi-level attention mechanisms represents a strategic layering of text information into structured ontological knowledge. The advent of the Transformer model further enhanced the capability to capture textual information, implicitly including ontological knowledge that substantially affects the performance of RE tasks. These innovations in attention structures, specifically designed to meet the demands of RE tasks for ontological knowledge learning, have revolutionized the approach and effectiveness.

With the advent of PLMs, the research landscape in nearly all tasks of NLP has undergone a significant transformation. These models have brought an unprecedented level of

semantic memory and understanding. Under the framework of self-supervised learning, PLMs understand the contextual semantics via massive corpora, subsequently undergoing task-specific fine-tuning. Nevertheless, fine-tuning often requires a substantial amount of task-specific data to adjust the entire model's parameters, which can lead to overfitting and a huge gap between the task's objective and the self-supervised learning objective. To address these limitations, prompt-tuning is proposed, offering a more efficient way to fine-tune PLMs with slight parameter adjustments. The prompt-tuning enhances task alignment with the objectives of self-supervised learning while also eliciting task-specific information from PLMs. These advantages both improve the models' application to specific tasks and mitigate the issues associated with traditional fine-tuning. Thus, how to construct effective, task-specific prompts to elicit specific knowledge embedded in these large models became a pivot problem. For RE tasks, the challenge lies in converting the semantic information mastered by PLMs into ontological knowledge through well-crafted prompts. This conversion aims to enhance the performance of these models across various application dimensions in RE tasks, ensuring a comprehensive and nuanced understanding and handling of relations.

2.1 Basic Ontological Knowledge Learning

In this section, we will discuss a series of representative research works, with a specific focus on how leveraging the advantages of ontological knowledge learning can address challenges within RE. Through this review, we aim to provide a comprehensive understanding of existing methodologies while emphasizing their contributions and limitations in tackling the recognized challenges in RE.

2.1.1 Piecewise Convolutional Neural Network

Given a bag of sentences $\mathcal{B} = \{s_1, \ldots, s_N\}$ containing a pair of subject $e^{(s)}$ and object $e^{(o)}$ entities, the distant supervision [34] assigns a relation label r to the sentence

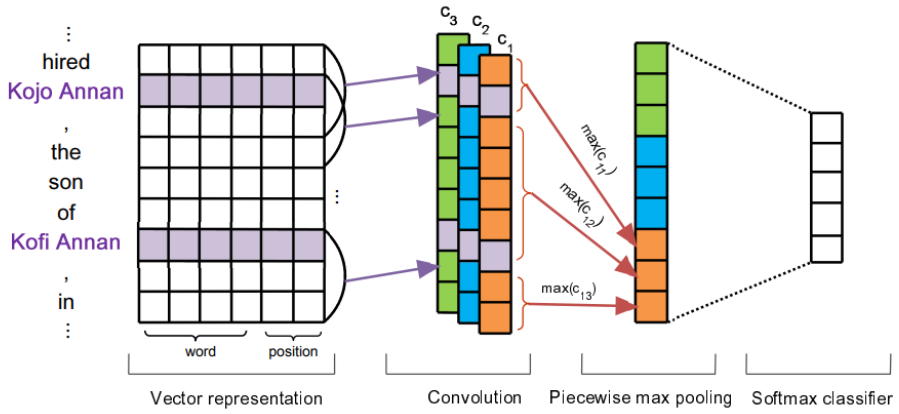

Figure 2.1 : The architecture of Piecewise Convolution Neural Network. Copied from original paper [1].

bag $\mathcal{B}$ based on the corresponding KG triple fact. The objective of relation extraction is to predict the relation label $\hat{r}$ to the sentence bag $\mathcal{B}$ for a given entity pair based on the sentence bag $\mathcal{B}$.

Word features Existing approaches derive a latent representation for each sentence s_j in the bag $\mathcal{B} = \{s_1, \ldots, s_N\}$ using word embedding [25]. In the following discussion, we omit the sentence index j for clarity. First the sentence s is tokenized into a sequence of n words, denoted as $s = [w_1, \ldots, w_l]$, where l is the length of the sentence. Then a word2vec method [25] is used to transform the discrete tokens into low-dimensional, real-valued embeddings, resulting in $s = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_l] \in \mathbb{R}^{d_w \times l}$, where $\boldsymbol{v}_i$ represents the word embedding of the i-th word in the sentence, and d_w denotes the dimension of the word embeddings.

Position features Traditionally, in tasks sensitive to ontological knowledge, structural features have been pivotal, as relying solely on word features proves insufficient to capture such complex information. Thus, position features are combined with word embeddings to specify the relative distances between each word w_i and the target entities $e^{(s)}$ and

$e^{(o)}$ [21]. These position features provide information about the positions of the entities within each sentence. For example, consider the sentence "Kojo Annan, the son of Kofi Annan...". The relative distances from the word "son" to the subject entity $e^{(s)}$ (*Kojo Annan*) are 3 and -2 words, respectively. Each relative distance is transformed into a real-valued vector by looking up the position embedding matrix for the subject and object entities, denoted as $\boldsymbol{x}_i^{(ps)}$ and $\boldsymbol{x}_i^{(po)} \in \mathbb{R}^{d_p}$, respectively. The final word vector representations will be represented as the concatenation of word embedding and position features, i.e., $\boldsymbol{x}_i^{(p)} = [\boldsymbol{v}_i; \boldsymbol{x}_i^{(ps)}; \boldsymbol{x}_i^{(po)}]$. $[;]$ denotes the operation of vector concatenation.

A common practice in distant supervision relation extraction for extracting ontological knowledge is to use a piecewise convolutional neural network (PCNN) [1] to generate contextualized representations over a sequence of word embeddings. Compared to the typical 1D-CNN with max-pooling [21], piecewise max-pooling can capture the structure information between two entities by considering their positions. First, a 1D-CNN [55] is applied to the input sequence of word embeddings $\boldsymbol{v}$, resulting in contextualized representations. Next, a piecewise max-pooling operation is performed over the output sequence to obtain sentence-level embedding. These steps are written as

$$\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n] = \text{1D-CNN}(\boldsymbol{V}; \theta^{(cnn)}), \tag{2.1}$$

$$\boldsymbol{s} = \tanh([Pool(\boldsymbol{H}^{(1)}); Pool(\boldsymbol{H}^{(2)}); Pool(\boldsymbol{H}^{(3)})]) \tag{2.2}$$

where $\boldsymbol{W}^{(c)} \in \mathbb{R}^{d_c \times Q \times d_x}$ is a conv kernel with window size of Q. $\boldsymbol{H}^{(1)}$, $\boldsymbol{H}^{(2)}$ and $\boldsymbol{H}^{(3)}$ are three consecutive parts of $\boldsymbol{H}$ by dividing $\boldsymbol{H}$ w.r.t. the indices of subject $e^{(s)}$ and object $e^{(o)}$ entities. Consequently, $\boldsymbol{s} \in \mathbb{R}^{d_h}$, is the resulting sentence-level embedding. The whole architecture of the PCNN is illustrated in Fig 2.1.

To alleviate the wrong labeling problem, the PCNN model employs the MIL framework. In this framework, we consider a set of bags $\{M_1, M_2, \ldots, M_T\}$, where each bag M_i contains multiple instances with the same pair of entities $M_i = \{m_i^1, m_i^2, \ldots, m_i^n\}$. The goal of the MIL framework is to predict the relationship label at the bag level for

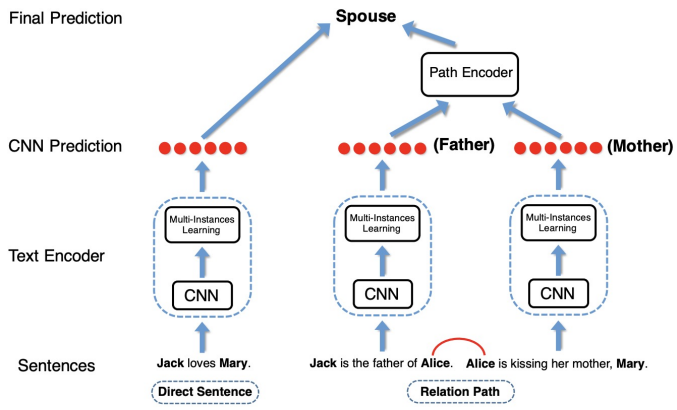

Figure 2.2 : The architecture of our neural relation extraction model with relation paths. Copied from original paper [2].

unseen bags. Given the training bags (M_i, y_i), where y_i represents the true relationship label for bag M_i, the PCNN model with parameters θ outputs a vector $\boldsymbol{o}$, where the r-th component $\boldsymbol{o}_r$ corresponds the score associated with relation r. To obtain the conditional probability $\sum_{j=1}^{n} p(r|m_i^j, \theta)$, a softmax operation is applied over all relations. The objective function is defined as follows, aiming to maximize the log probability of the true relationship label for each bag:

$$J(\theta) = \sum_{i=1}^{T} \log p(y_i|m_i^j; \theta) \tag{2.3}$$

2.1.2 Incorporating Relation Path

Normally, existing models solely rely on those direct sentences containing both entities. To further capture the ontological knowledge learning, a path-based method [2] is proposed to encode the relational semantics from both direct sentences and inference chains. There are also many sentences containing only one of the target entities, which also provides rich useful information but not yet fully employed. Relation path encoder measures the probability of each relation r given a relation path in the text. This will utilize the inference chain structure to help make predictions. As illustrated in Fig 2.2, a

convolution neural network (CNN) first embeds the semantics of sentences. Afterward, a relation path is built for measuring the probability of relations given an inference chain in the text. Finally, the direct sentences and relation paths are concatenated to predict the relations.

More specifically, a path p_1 is defined between (h, t) as $(h, e), (e, t)$, and the corresponding relations are r_A, r_B. Each of (h, e) and (e, t) corresponds to at least one sentence in the text. The probability of relation r conditioned on p_1 as follows,

$$p(r|r_A, r_B) = \frac{\exp(o_r)}{\sum_{i=1}^{n_r} \exp(o_i)}, \tag{2.4}$$

where o_i measures how well relation r matches with the relation path (r_A, r_B). Followed with [37], each relation r will be initialized with a unique, distributed representation. Then, the similarity is calculated as follows:

$$o_i = -\|r_i - (r_A + r_B)\|_{L_1}. \tag{2.5}$$

Here, there is an implicit assumption that if r_i is semantically similar to relation path $p_i : h \xrightarrow{r_A} e \xrightarrow{r_B} t$ and the embedding r_i will be closer to the relation embedding $(\boldsymbol{r}_A + \boldsymbol{r}_B)$. Finally, the relation-path score function is shown as follows:

$$G(h, r, t|\pi) = E(h, r_A, e)E(e, r_B, t)p(r|r_A, r_B), \tag{2.6}$$

where $E(h, r_A, e)$ and $E(h, r_B, t)$ measure the probabilities of relational facts (h, r_A, e) and (e, r_B, t) from text, and $p(r|r_A, r_B)$ measures the probability of relation r given relation path (r_A, r_B). In summary, it is evident that as our understanding of tasks sensitive to textual data and ontological knowledge deepens, the specially designed models and methodologies play a crucial role in enhancing the model's capability to capture and comprehend ontological knowledge. The intricacies of these designs not only reflect a heightened level of cognitive sophistication but also underscore the significance of tailored approaches in dealing with complex data structures and knowledge domains.

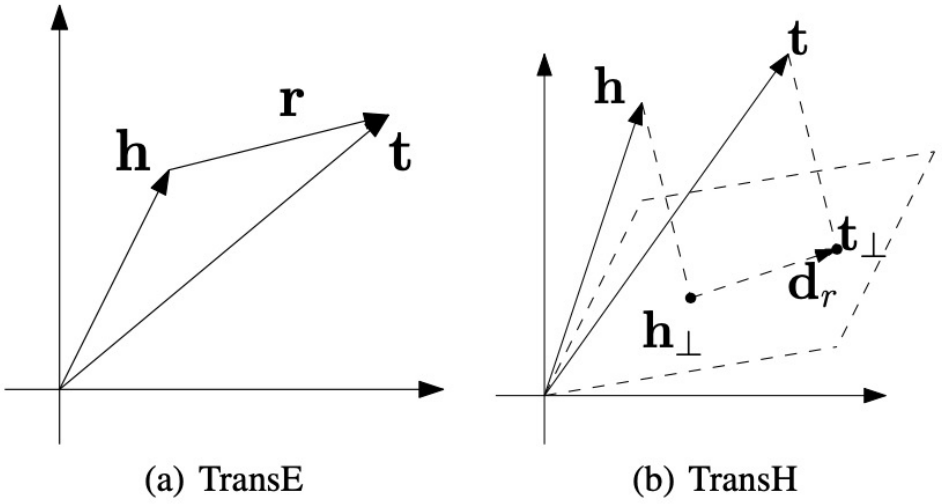

Figure 2.3 : Simple illustration of TransE and TransH. Copied from original paper [3].

2.1.3 Knowledge Graph Embedding via Translation

The acquisition and representation of ontological knowledge remain central challenges in the field of semantic computing and artificial intelligence. KGs have emerged as a pivotal structure for organizing ontological knowledge, enabling machines to understand. Among the various strategies for KG representation, embedding models, particularly those utilizing translation-based approaches, have shown great promise.

We delve into the specifics of how these models not only learn to encode but also to interpret and utilize ontological knowledge. Models such as TransE [56] and TransH [3] serve as representations of this approach, leveraging the concept of translation in a multi-dimensional space to establish relational links between entities. These models offer a geometric perspective to intuitively and effectively capture the complex hierarchies and interdependencies in KGs.

TransE represents a relation by translation vector r so that the pair of embedded entities in a triplet (h, r, t) can be connected by r with very low error. Although it is highly ef-

ficient, TransE struggled to deal with reflexive/one-to-many/many-to-one/many-to-many relations. Formally, TransE models a relation r as follows:

$$e = \|\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}\|_{\ell_{1/2}}, \quad r \in \mathbb{R}^k \tag{2.7}$$

where e represents the error of this triplet (h, r, t). The error will be low if (h, r, t) is a golden triplet.

TransH allows for an entity to possess diverse representations contingent upon its engagement with distinct relations. Thus, it can overcome the problems of TransE in modeling reflexive/one-to-many/many-to-one/many-to-many relations. As shown in Fig 2.3, for a relation r, the relation-specific translation vector $\boldsymbol{d}_r$ is positioned in the relation-specific hyperplane $\boldsymbol{w}_r$ (the normal vector) rather than in the same space of entity embeddings. The embedding $\boldsymbol{h}$ and $\boldsymbol{t}$ are first projected to the hyperplane $\boldsymbol{w}_r$. The projections are denoted as $\boldsymbol{h}_\perp$ and $\boldsymbol{t}_\perp$. Further, the authors assume $\boldsymbol{h}_\perp$ and $\boldsymbol{t}_\perp$ can be connected by a translation vector $\boldsymbol{d}_r$ on the hyperplane with a low error if $(h, r, t)$is a golden triplet. Thus, a scoring function is formulated as follows:

$$\begin{aligned}
\boldsymbol{h}_\perp &= \boldsymbol{h} - \boldsymbol{w}_r^T \boldsymbol{h} \boldsymbol{w}_r, \\
\boldsymbol{t}_\perp &= \boldsymbol{t} - \boldsymbol{w}_r^T \boldsymbol{t} \boldsymbol{w}_r, \\
f_r(\boldsymbol{h}, \boldsymbol{t}) &= \left\| (\boldsymbol{h} - \boldsymbol{w}_r^T \boldsymbol{h} \boldsymbol{w}_r) + \boldsymbol{d}_r - (\boldsymbol{t} - \boldsymbol{w}_r^T \boldsymbol{t} \boldsymbol{w}_r) \right\|_{\ell_2}^2 .
\end{aligned}$$

2.2 Attention Mechanism for Ontological Knowledge Learning

Before delving deeper into ontological knowledge learning, it is crucial to introduce a mechanism that has nearly reshaped the feature-capturing process in almost all NLP tasks: the attention mechanism [57]. In this section, we aim to explore how the attention mechanism, a pivotal innovation in the field, has transformed our approach to understanding and processing language, particularly in enhancing the acquisition and application of ontological knowledge in various linguistic tasks.

2.2.1 Vanilla Attention mechanism

The intuition behind the attention mechanism is rooted in the way humans focus on different parts of information at different times to comprehend and interpret the world around them. Essentially, the attention mechanism provides a way for models to allocate their computational focus in a manner akin to human attention, leading to a more nuanced and effective understanding and processing of data. The vanilla attention mechanism takes three inputs: keys, values, and queries. Each of them consists of a set of representations. The Queries are denoted as $\boldsymbol{Q} = [\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_n] \in \mathbb{R}^{d_q \times n}$. Keys are denoted as $\boldsymbol{K} = [\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots, \boldsymbol{k}_n] \in \mathbb{R}^{d_k \times n}$, and its corresponding values are denoted as $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n] \in \mathbb{R}^{d_v \times n}$. Commonly, the keys $\boldsymbol{K}$ and the values $\boldsymbol{V}$ are often derived from the same source. The attention mechanism calculates the alignment score between $\boldsymbol{k}_i$ and $\boldsymbol{q}_i$ by a function $f(\boldsymbol{k}_i, \boldsymbol{q}_j)$. The alignment score can capture dependencies, relevance, or similarity between the keys and queries. A softmax function is applied to obtain a categorical distribution indicating the importance of each value to the corresponding query. Concretely, a $p(z = i|\boldsymbol{K}, \boldsymbol{q}_j)$ represents $\boldsymbol{v}_i$ could contribute crucial information to $\boldsymbol{q}_i$:

$$p(z = i|\boldsymbol{K}, \boldsymbol{q}_j) = \frac{\exp(f(\boldsymbol{k}_i, \boldsymbol{q}_j))}{\sum_{i=l}^{n} \exp(f(\boldsymbol{k}_l, \boldsymbol{q}_j))} \tag{2.8}$$

where z is a random variable indicating which token is important to q_j for a specific task.

It is worth noting that there are different choices for the specific form of the function $f(\boldsymbol{k}_i, \boldsymbol{q}_j)$, such as additive, multiplicative, and scaled dot-product attention mechanisms. The additive attention mechanism is defined as follows:

$$f(\boldsymbol{k}_i, \boldsymbol{q}_j) = \boldsymbol{w}^T \sigma(\boldsymbol{W}^{(1)} \boldsymbol{k}_i + \boldsymbol{W}^{(2)} \boldsymbol{q}_j) \tag{2.9}$$

where $\boldsymbol{W}^{(1)} \in \mathbb{R}^{d_h \times d_e}$, $\boldsymbol{W}^{(2)} \in \mathbb{R}^{d_h \times d_e}$ and $\boldsymbol{w} \in \mathbb{R}^{d_h}$ are the learnable parameters. $\sigma(\cdot)$ denotes the activate function. The multiplicative attention mechanism is defined as:

$$f(\boldsymbol{k}_i, \boldsymbol{q}_j) = \langle \boldsymbol{W}^{(1)} \boldsymbol{k}_i, \boldsymbol{W}^{(2)} \boldsymbol{q}_j \rangle \tag{2.10}$$

where $\langle \cdot \rangle$ could denote inter-product or cosine similarity. The scaled dot-product attention mechanism is defined as:

$$\boldsymbol{S} = \boldsymbol{V}\mathrm{softmax}(\frac{\boldsymbol{Q}^T\boldsymbol{K}}{\sqrt{d_q}})^T \tag{2.11}$$

where d_q represents the dimension of the query. It is a scale factor to make the distribution smoother, avoid gradient vanishing and constrain the range of variance.

2.2.2 Selective Attention Mechanism

To address the issue of wrongly labeled sentences, the selective attention mechanism [37] is proposed to de-emphasize the impact of noisy sentences and extract the ontological knowledge from the sentences that contain the same entity pair. This selective attention mechanism assigns attention weights to each sentence in a bag based on its relevance to the predicted relation. This allows the model to focus more on informative sentences and ignore the noisy ones. The attention weight α_i for each sentence $\boldsymbol{x}_i$ is calculated using a softmax function:

$$\alpha_i = \frac{\exp(\boldsymbol{e}_i)}{\sum_k \exp(\boldsymbol{e}_k)} \tag{2.12}$$

where $\boldsymbol{e}_i$ is defined as a query-based function that measures the matching degree between sentence $\boldsymbol{x}_i$ and then predicts relation $\boldsymbol{r}$ in a distributed representation. Specifically, $\boldsymbol{e}_i$ is obtained through a bilinear form:

$$\boldsymbol{e}_i = \boldsymbol{x}_i\boldsymbol{A}\boldsymbol{r} \tag{2.13}$$

Finally, the bag representation $\boldsymbol{o}_j$ is computed as the weighted sum of the sentence representations:

$$\boldsymbol{o}_j = \sum_i \alpha_i \boldsymbol{x}_i \tag{2.14}$$

By assigning attention weights to each sentence based on its relevance to the predicted relation, the selective attention network can effectively filter out the noise and focus on the informative sentences for relation extraction.

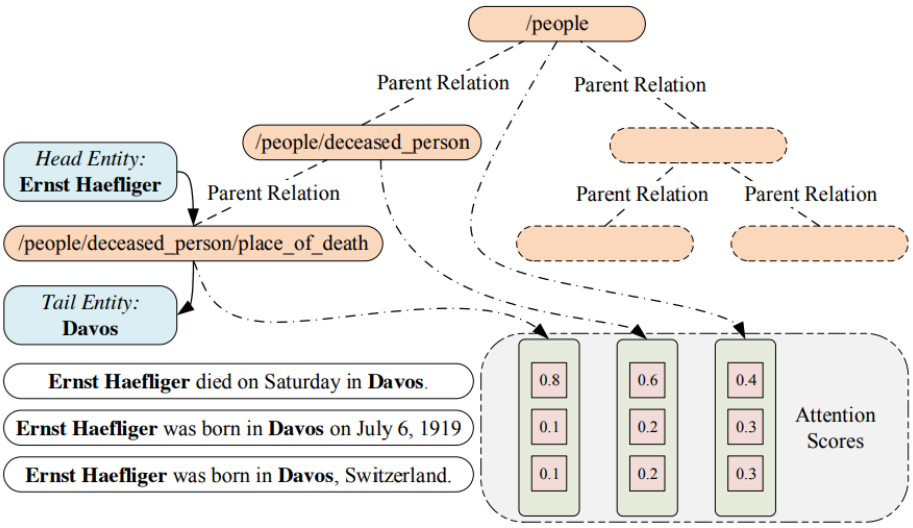

Figure 2.4 : The sample of hierarchical attention mechanism in DSRE. Copied from original paper [4].

2.2.3 Hierarchical Attention Mechanism

In the context of ontological knowledge learning, a fundamental challenge lies in accurately capturing and utilizing the structured information inherent in textual data. This is particularly crucial in tasks such as Relation Extraction (RE), where the objective is subtly aligned with understanding and predicting the relationships embedded within the text. Rather than directly predicting the relation label $\hat{r}$ of an entity pair from a corresponding sentences' bag $\mathcal{B}$, the focus shifts towards leveraging the hierarchical nature of labels to further extract the ontological knowledge from the text. Coarsed-grained relation labels, denoted as $[r^{(1)}, \ldots, r^{(M)}]$, can be derived from the mention of r. For example, if $r =$ /BUSINESS/COMPANY/FOUNDERS. $r^{(1)} =$ /BUSINESS/COMPANY and $r^{(2)} =$ /BUSINESS signify broader categorical relationships. By integrating this hierarchical label information with attention mechanisms, a hierarchical attention framework can be constructed, which is illustrated in Fig 2.4. This approach allows for a more nuanced and layered extraction of ontological knowledge from the text, aligning closely with the objectives of

ontological knowledge learning and providing a more robust foundation for tasks like RE. In the hierarchical attention mechanism, attention scores are calculated for each instance based on their significance in expressing the corresponding relation. The mechanism can be defined as follows:

$$\boldsymbol{o}_j^{(l)} = \sum_i \alpha_i \boldsymbol{x}_i, \tag{2.15}$$

$$\boldsymbol{e}_i^{(l)} = \boldsymbol{q}_{r^l}^T \boldsymbol{W}_s \boldsymbol{x}_i, \tag{2.16}$$

$$\alpha_i^{(l)} = \frac{\exp(\boldsymbol{e}_i^{(l)})}{\sum_k \exp(\boldsymbol{e}_k^{(l)})}, \tag{2.17}$$

where $\boldsymbol{q}_r$ is a query vector corresponding to each grained relation $r \in \mathcal{R}$ and $\boldsymbol{W}_s$ is the weight matrix. Finally, the sentence representations generated by different-grained relations are concatenated together to form the final representation $\boldsymbol{o}$.

$$\boldsymbol{o} = [\boldsymbol{o}^{(0)}; \boldsymbol{o}^{(1)}; \ldots; \boldsymbol{o}^{(M)}] \tag{2.18}$$

By incorporating the hierarchical attention mechanism, the model can effectively capture the hierarchical structure of relations and generate informative representations.

2.2.4 Knowledge Graph-enhanced Attention Mechanism

In the diverse landscape of NLP, certain tasks are particularly sensitive to ontological knowledge, necessitating a way to extract comprehensive ontological knowledge from the textual data or hierarchical labels. However, these tasks often grapple with the challenge of long-tail distributions, where numerous categories are scarcely represented, making it difficult for traditional models to perform effectively. Furthermore, the ontological knowledge constrained in text information is limited after all.

To address these complexities, the integration of Knowledge Graphs and Attention Mechanisms has emerged as a powerful strategy [5], enhancing the sensitivity and adaptability of models to ontological nuances. It introduces a sophisticated approach that leverages Knowledge Graph Embeddings and Graph Convolution Networks, coupled with a

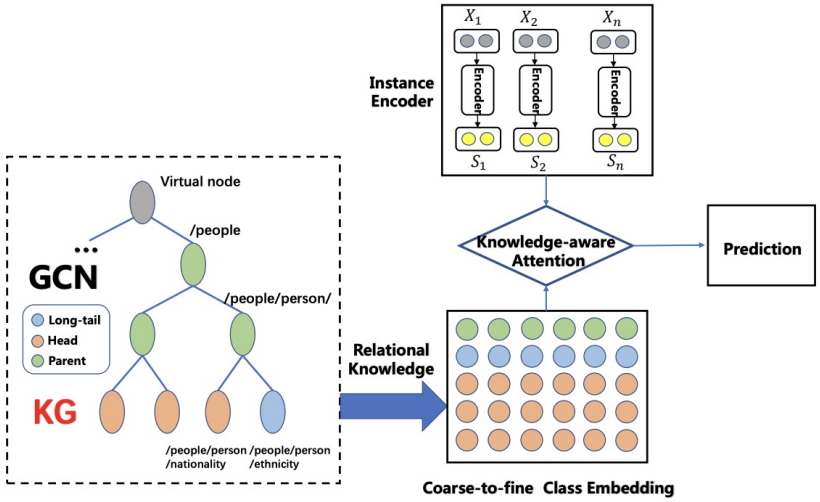

Figure 2.5 : The architecture of Knowledge Graph Embeddings and Graph Convolution Networks. Copied from original paper [5].

coarse-to-fine knowledge-aware attention mechanism. This method significantly boosts performance by deeply integrating and utilizing ontological knowledge, offering insights and advancements in handling tasks sensitive to the rich and varied nature of ontological structures. We will explore the innovative methodology, the results it yields, and its broader contributions and potential impact on tasks that require a nuanced understanding of ontological knowledge.

For each instance $s = w_1, \ldots, w_n$ with two mentioning entities, zhang et al., 2019 [5] encode the raw instance into a continuous low-dimensional vector $\boldsymbol{x}$, which consists of an embedding layer [1], and an encoder layer [1, 21]. Then, given pre-trained KG embeddings and predefined relation hierarchies, a two-layer GCNS [58] is leveraged to learn the explicit fine-grained relational knowledge from the label space.

Hierarchy Label Graph Construction. Given a relation set $\mathcal{R}$ of a KG $\mathcal{G}$ (e.g., Freebase), which consists of base-level relations, the corresponding higher-level relation set $\mathcal{R}^H$. According to the former mentioned, the coarse-grained relations are more general

and contain several different fine-grained relations in a tree structure. As shown in Figure 2.5, a virtual father node is used to construct the most coarse-grained relation associations between relations. The vectors of each node in the bottom layer are initialized through pretrained TransE [56] KG embeddings.

GCN Output Layer. GCN is applied to learn explicit relational knowledge among relations because the implicit relevant information obtained by KG embeddings for each relation is not enough. Formally, the label vectors of the fine-grained and coarse-grained for the i-th label to form,

$$v_i^1 = f\left(W^1 v_i + \sum_{j \in \mathcal{N}_p} \frac{W_p^1 v_j}{|\mathcal{N}_p|} + \sum_{j \in \mathcal{N}_c} \frac{W_c^1 v_j}{|\mathcal{N}_c|} + b_g^1\right) \tag{2.19}$$

where $W^1 \in \mathbb{R}^{q \times d}$, $W_p^1 \in \mathbb{R}^{q \times d}$, $W_c^1 \in \mathbb{R}^{q \times d}$, $b_g^1 \in \mathbb{R}^q$, f in the rectified linear unit [59] function, and $\mathcal{N}_c(\mathcal{N}_p)$ is the index set of the i-th labels fine-grained (coarse-grained). The second layer follows the same formulation as the first layer and outputs $v_i^{explicit}$. Finally, both pretrained $v_i^{implicit}$ with GCNs node vector $v_i^{explicit}$ are concatenated to form hierarchy class embeddings,

$$q_r = v_i^{implicit} || v_i^{explicit} \tag{2.20}$$

where $q_r \in \mathbb{R}^{d+q}$. For the following, the class embeddings containing useful ontological knowledge for long-tails among labels will be treated as a query for matching sentence vectors. Hence, the relation extraction problem becomes a retrieval problem.

2.3 Attention-Driven Language Models

In the preceding section, we delved into the intricacies of attention mechanisms and their integration with ontological knowledge learning, highlighting how these mechanisms enhance the sensitivity and adaptability of models to complex and hierarchical ontological knowledge inherent in textual data and relations. Building upon this foun-

dation, this section shifts focus toward a spectrum of language models that are fundamentally grounded in attention architecture. Attention-Driven Language Models have revolutionized the field of Natural Language Processing (NLP), offering a more nuanced and contextually aware approach to understanding textual data. From self-attention neural networks to their various evolutions, attention mechanisms have played a crucial role, not only enhancing the understanding of textual semantics but also, with the scaling of model sizes, fundamentally reshaping the entire NLP tasks.

2.3.1 Directional Self-Attention Network

Unlike conventional models that rely on RNN and CNN architectures, Directional Self-Attention Network (DiSAN) only consists of a directional self-attention with temporal order encoded followed by a multi-dimensional attention. The attention mechanism offers greater computational adaptability in handling sequence lengths compared to Recurrent Neural Networks (RNNs) or Convolutional Neural Networks (CNNs). It also provides a more task/data-driven approach to modeling dependencies. Differing from sequential models, attention-based computations can be significantly and efficiently accelerated using existing distributed or parallel computing frameworks.

Generally, a sentence is denoted by a sequence of discrete tokens $\boldsymbol{v} = [v_1, v_2, \ldots, v_n]$. A pre-trained token embedding (e.g., Word2Vec [25] or GloVe [24]) is applied to $\boldsymbol{v}$ and transforms all discrete token to a sequence a sequence of low-dimensional dense vector representations $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$ with $x_i \in \mathbb{R}^{d_e}$

In DiSAN, there are two major attention mechanisms: multi-dimensional attention and directional self-attention. Multi-dimensional attention is a neural extension of additive attention (or Multi-layer Perceptron Attention) at the feature level. When extending multi-dimension to self-attention, there are two variants of multi-dimensional attention. The first one, called multi-dimensional "**token2token**" self-attention, explores the dependency between x_i and x_j from the same source $\boldsymbol{x}$, and generates context-aware coding for

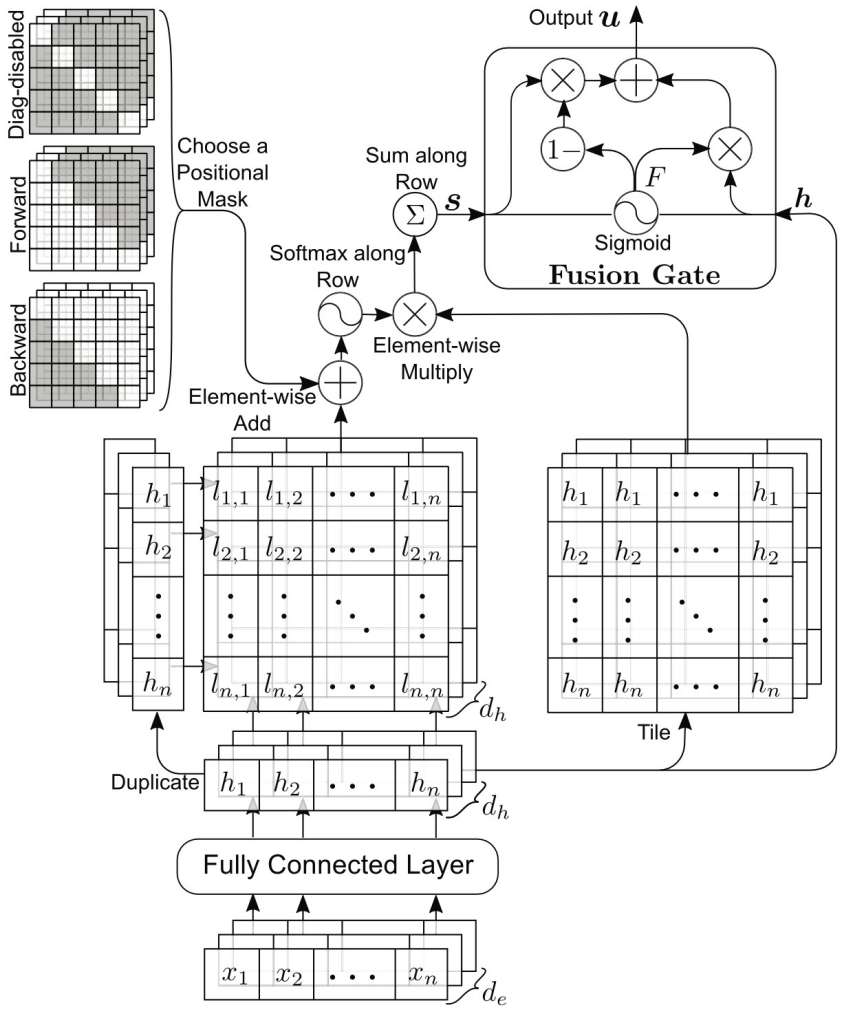

Figure 2.6 : The architecture of Directional self-attention (DiSA) mechanism. Here, $l_{i,j}$ denotes $f(h_i, h_j)$. Copied from original paper [6].

each element. It replaces q and x_j as the following:

$$f(x_i, x_j) = W^T \sigma\left(W^{(1)} x_i + W^{(2)} x_j + b^{(1)}\right) + b. \tag{2.21}$$

where $f(x_i, x_j) \in \mathbb{R}^{d_e}$ is a vector with the same length as x_i, and all the weight matrices $W, W^{(1)}, W^{(2)} \in \mathbb{R}^{d_e \times d_e}$. Then, a probability matrix $P^j \in \mathbb{R}^{d_e \times n}$ is calculated for each x_j as $P^j_{ki} \triangleq p(z_k = i | x, x_j)$. The output x_j is as shown:

$$s_j = \sum_{i=1}^{n} P^j_{\cdot i} \odot x_i. \tag{2.22}$$

The second one, multi-dimensional "**source2token**" self-attention, explores the dependency between x_i and the entire sequence $\boldsymbol{x}$, and compresses the sequence $\boldsymbol{x}$ into a vector. Unlike Eq. 2.21, q has been removed from the following formulation:

$$f(x_i) = W^T \sigma \left(W^{(1)} x_i + b^{(1)} \right) + b. \tag{2.23}$$

The probability matrix is defined as $P_{ki} \triangleq p(z_k = i|x)$ and is computed in the same way as P in vanilla multi-dimensional attention. The output s is also same, i.e.,

$$s = \sum_{i=1}^{n} P_{\cdot i} \odot x_i. \tag{2.24}$$

Based on these attentions, Directional Self-Attention Network (DiSAN) is proposed for sentence-encoding without any recurrent or convolutional structure.

2.3.2 Transformer

Since the introduction of recurrent language models and convolutional neural networks, there have been continuous efforts to push their boundaries [33, 60, 61]. The Transformer [7] is proposed as a self-attention network that entirely dispenses with recurrence and convolutions. It has become one of the most widely used models for natural language processing tasks. The Transformer architecture consists of an encoder and a decoder, making it suitable for various NLP tasks such as neural machine translation and sentiment analysis. The overall architecture of the Transformer is illustrated in Fig 2.7. In the following paragraphs, we will delve into the key components that constitute the Transformer, discussing each in detail.

Positional Encoding. Positional encoding is crucial for preserving positional information in sequences since the attention mechanism is permutation-intensitive. It is defined as follows:

$$PE(pos, 2i) = sin(pos/10000^{2i/d}) PE(pos, 2i+1) = cos(pos/10000^{2i/d}) \tag{2.25}$$

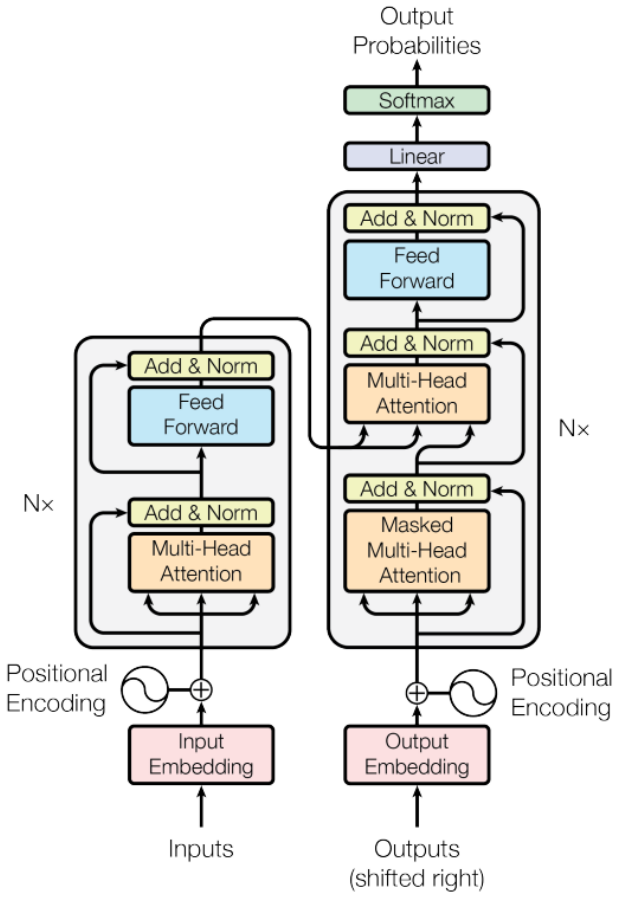

Figure 2.7 : The architecture of Transformer. Copied from original paper [7].

where pos is the position and i is the dimension. Each dimension of the positional encoding corresponds to a sinusoid.

Multi-Head Attention. The Multi-Head Attention mechanism enhances the diversity of attention by using multiple sets of key-value-query attention mechanisms. It is defined as follows:

$$\boldsymbol{S} = \text{MultiHeadAttn}(\boldsymbol{K}, \boldsymbol{V}, \boldsymbol{Q}) = \boldsymbol{W}[\boldsymbol{H}_1; \ldots; \boldsymbol{H}_h], \tag{2.26}$$

$$\boldsymbol{H}_c = \text{ScaleDotProdAttn}(\boldsymbol{W}_c^{(k)}\boldsymbol{K}, \boldsymbol{W}_c^{(v)}\boldsymbol{V}, \boldsymbol{W}_c^{(q)}\boldsymbol{Q}) \tag{2.27}$$

where $W^{(\cdot)}$ denotes the learnable matrix.

Masked Multi-Head Attention. This component is specifically designed for the Transformer decoder. Without the masked operation, the model may have access to future tokens during the decoding process. To prevent this, a mask matrix is applied, allowing attention only between $\boldsymbol{k}_i$ if their corresponding positions satisfy certain conditions.

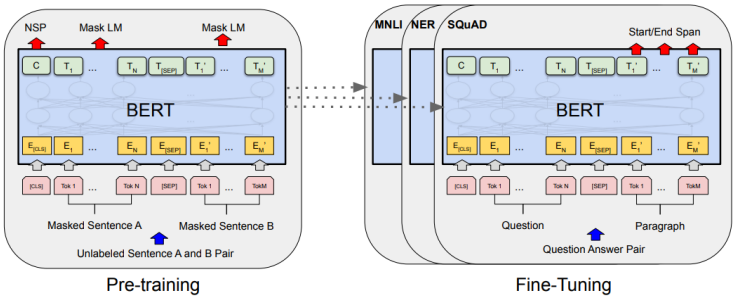

Figure 2.8 : The architecture of BERT. Copied from original paper [8].

Transformer Encoder. The sentences are first embedded with additive positional encoding. Then, each representation is passed through multiple layers of the multi-head attention mechanism. Each layer consists of a feed-forward network with an activation function and includes residual connection [62] and layer normalization [63]. The procedure can be summarized as follows:

$$
\begin{aligned}
\boldsymbol{H} &= [h_1, \ldots, h_n] \triangleq \boldsymbol{X}^{(N)} \in \mathbb{R}^{d_w \times n}, \\
\boldsymbol{X}^{(l+1)} &= \mathrm{LayerNorm}(\mathrm{FFN}(\boldsymbol{X}^{(l+1)\prime}) + \boldsymbol{X}^{(l+1)\prime}), \\
\boldsymbol{X}^{(l+1)} &= \mathrm{LayerNorm}(\mathrm{MultiHeadAttn}(\boldsymbol{X}^{(l)}, \boldsymbol{X}^{(l)}, \boldsymbol{X}^{(l)}) + \boldsymbol{X}^{(l)}), \\
\boldsymbol{X}^{(0)} &= \boldsymbol{X} + \boldsymbol{W}^{(pe)},
\end{aligned}
$$

where H is a sequence of distributed representation, $\boldsymbol{W}^{(pe)} \in \mathbb{R}^{d_e \times n}$ is a learnable weights of position embedding.

Transformer Decoder. This process is briefly denoted as:

$$
\begin{aligned}
&\boldsymbol{S} = [s_1, \ldots, s_n] \triangleq \boldsymbol{Z}^{(N)} \in \mathbb{R}^{d_w \times n}, \\
&\boldsymbol{Z}^{(l+1)} = \mathrm{LayerNorm}(\mathrm{FFN}(\boldsymbol{Z}^{(l+1)\prime}) + \boldsymbol{Z}^{(l+1)\prime}), \\
&\boldsymbol{Z}^{(l+1)\prime} = \mathrm{LayerNorm}(\text{Masked-MultiHeadAttn}(\boldsymbol{H}, \boldsymbol{H}, \boldsymbol{Z}^{(l+1)\prime\prime}) + \boldsymbol{Z}^{(l+1)\prime\prime}), \\
&\boldsymbol{Z}^{(l+1)\prime\prime} = \mathrm{LayerNorm}(\text{Masked-MultiHeadAttn}(\boldsymbol{Z}^{(l)}, \boldsymbol{Z}^{(l)}, \boldsymbol{Z}^{(l)}) + \boldsymbol{Z}^{(l)}), \\
&\boldsymbol{Z}^{(0)} = \boldsymbol{Z} + \boldsymbol{W}^{(pe)},
\end{aligned}
$$

where $\boldsymbol{S}$ is a sequence of decoding hidden states, $\boldsymbol{H}$ is the resulting hidden states, Z represents the left-shifted token list. The LayerNorm, Feed-Forward Network (FFN), and Masked Multi-Head Attention components have similar definitions as in the Transformer Encoder. The Transformer decoder takes the encoded input sequence and generates the output sequence token by token. It utilizes self-attention mechanisms to attend to different parts of the input sequence while making predictions. The positional encoding ensures that the model considers the order of the tokens in the input. By stacking multiple layers of self-attention and feed-forward networks, the Transformer decoder can capture complex dependencies and generate high-quality translations or predictions.

2.3.3 Pre-trained Language Models: GPT and BERT

PLMs have revolutionized the field of NLP by learning contextualized representations from large-scale corpora. Two prominent pre-trained models, Generative Pre-trained Transformer (GPT) [39, 41] and Bidirectional Encoder Representations from Transformers (BERT) [8], have achieved remarkable performance on various NLP tasks.

GPT, based on the Transformer architecture, is a generative model that predicts the next word in a sentence given the previous context. It is trained using an auto-regressive objective, where the model is conditioned on the left context to generate the next word. GPT excels at tasks that require generating coherent and contextually relevant text, such as text completion and language translation.

On the other hand, BERT, also based on the Transformer architecture, is a discriminative model that learns bidirectional representations of words. It is trained using a masked language model objective, where random tokens in the input are masked, and the model is tasked with predicting the masked tokens based on the surrounding context. BERT captures contextual dependencies in both directions, allowing it to understand the relationships between words and their contexts. It performs exceptionally well on tasks such as question answering, named entity, and sentiment analysis. The architecture of BERT is shown in Figure 2.8.

The main difference between GPT and BERT lies in the training objectives. GPT is trained to generate coherent text, while BERT focuses on learning deep contextualized representations by predicting masked tokens. This difference in training objectives leads to variations in their capabilities and strengths. GPT is more suitable for tasks involving text generation, while BERT excels at tasks requiring a deep understanding of contextual information and semantic relationships.

Both GPT and BERT have made significant contributions to NLP, and their pre-trained representations can be fine-tuned for specific downstream tasks, providing a powerful foundation for various natural language understanding and generation tasks.

2.3.4 Prompt Tuning with Rules

Prompt tuning [46,47,49,50,51,52,53,54] is proposed as an approach to bridge the gap between the pre-training and fine-tuning objectives by introducing task-specific prompts. During pre-training, which typically involves self-supervised tasks like masked language modeling, the model predicts the marked word in a sentence. Prompt tuning leverages this concept and formulates prompts in a cloze-style format, where certain words are masked and the model is trained to predict them based on the surrounding context. The prompt tuning approach consists of a template and a set of label words. For instance, in natural language inference, the label words may be “yes”, “maybe”, and “no” cor-

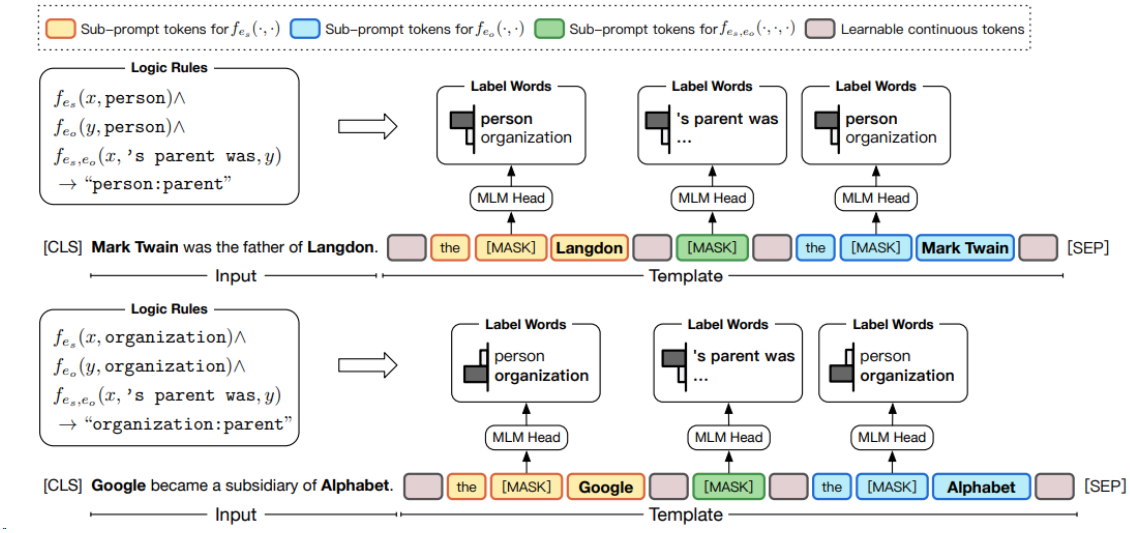

Figure 2.9 : The logic rule samples of PTR. Copied from original paper [9].

responding with {"entailment", "neutral", "contradiction"}, while in binary sentiment classification, the label words may be "good" and "bad" corresponding with the positive sentiment and the negative sentiment. However, designing effective prompts for the RE task is challenging due to the semantic overlap between relations. To address this challenge, some approaches [50, 51, 54] automatically generate prompts or replace discrete prompt tokens with continuous vectors. While these methods achieve reasonable performance, they often fall short compared to manually designed prompts. Manual Prompt designing allows for the incorporation of ontological knowledge, which is critical for RE tasks. To leverage prompt-tuning for RE, a recent approach called Prompt Tuning with Rules (PTR) [9] is introduced. Instead of directly designing task-specific prompt templates, PTR designs several sub-prompts and combines them using logic rules to form final task-specific prompts. The key distinction is that the logic rules in PTR are encoded with prior knowledge, specifically related to the semantics and ontological knowledge of named entities. For example, as shown in Figure 2.9, PTR utilizes logic rules to construct prompts for the PERSON:PARENT and ORGANIZATION:PARENT relations. The prompts consist of sub-prompts that determine the type of named entities and the semantics of the

sentence. However, designing multiple sub-prompts for each relation and finding valid logic rules to combine them can be time-consuming and labor-intensive. Additionally, with the increasing number of relations, this approach becomes less scalable. Furthermore, how to effectively provide ontological knowledge to stimulate the corresponding knowledge in PLMs for RE tasks in data-scarce scenarios remains an open question.

Chapter 3

Self-Attention Enhanced Selective Gate with Entity-Aware Embedding

3.1 Introduction

To alleviate the noisy labeling problem, Riedel et al. [35] proposes a multi-instance learning framework, which relaxes the strong assumption to *expressed-at-least-one* assumption. In plainer terms, this means any possible relation between two entities holds true in at least one distantly-labeled sentence rather than all of them that contain those two entities. In particular, instead of generating a sentence-level label, this framework assigns a label to a *bag* of sentences containing a common entity pair, and the label is a relationship between the entity pair on KGs. Recently, based on the labeled data at bag level, a line of works [1, 4, 11, 37, 64] under selective attention framework [37] let model implicitly focus on the correctly labeled sentence(s) by an attention mechanism and thus learn a stable and robust model from the noisy data.

However, the selective attention framework is vulnerable to situations where a bag is merely comprised of one single sentence labeled. And what is worse, only one sentence possibly expresses inconsistent meaning with the bag-level label. This scenario is not uncommon. For a popular distant supervised relation extraction benchmark, e.g., NYT dataset [35], up to 80% of its training examples (i.e., bags) are one-sentence bags. From our data inspection, we randomly sample 100 one-sentence bags and find 35% of them is incorrectly labeled. Two examples are shown in Table 3.1. These results indicate that in the training phase, the selective attention module is enforced to output a single-valued scalar for nearly 80% samples, leading to an ill-trained attention module.

Bag consisting of one sentence	**Label**	**Correct**
After moving back to *New York*, *Miriam* was the victim of a seemingly racially motivated attack ...	place_lived	True
... he faced, walking *Bill Mueller* and giving up singles to Mark Bellhorn and *Johnny Damon*.	place_lived	False

Table 3.1 : Two examples of one-sentence bags, which are correctly and wrongly labeled by distant supervision respectively.

Motivated by the aforementioned observations, we propose a novel **Se**lective **G**ate (SeG) framework for distantly supervised relation extraction. In the proposed framework, 1) we employ both the entity embeddings and relative position embeddings [21] for relation extraction, and an entity-aware embedding approach is proposed to dynamically integrate entity information into each word embedding, yielding more expressively powerful representations for downstream modules; 2) to strengthen the capability of widely-used piecewise CNN (PCNN) [1] on capturing long-term dependency [65], we develop a light-weight self-attention [6, 66] mechanism to capture rich dependency information and consequently enhance the capability of the neural network via producing complementary representation for PCNN; and 3) based on the preceding versatile features, we design a selective gate to aggregate sentence-level representations into bag-level ones and alleviate intrinsic issues appearing in selective attention.

Compared to the baseline framework (i.e., selective attention for multi-instance learning), SeG is able to produce entity-aware embeddings and rich-contextual representations to facilitate downstream aggregation modules that learn from noisy training data. Moreover, SeG uses the gate mechanism with pooling to overcome the problem occurring in selective attention, which is caused by one-sentence bags. In addition, it still keeps the

light-weight structure to ensure the scalability of this model.

3.2 Approach

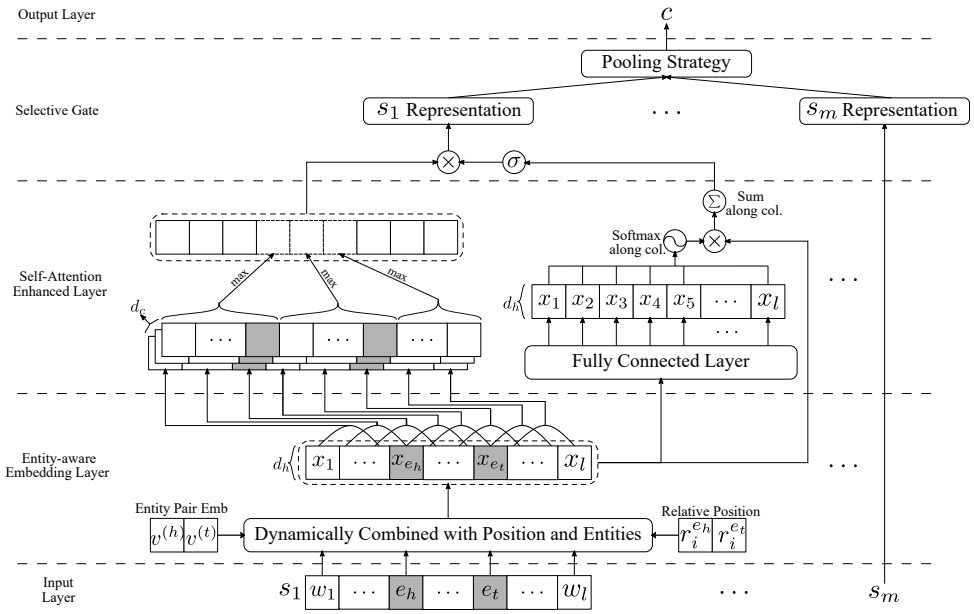

Figure 3.1 : The framework of our novel model without the pooling strategy used for sentence encoder has two main components: (1)Entity-Aware Embedding (2)Self-Attention Enhanced Selective Gate. As an example, tokens e_h^k and e_t^k in the gray background mean the head entity and tail entity of this sentence.

As illustrated in Figure 3.1, we propose a novel neural network, i.e., SeG, for distantly supervised relation extraction, which is composed of the following neural components.

3.2.1 Entity-Aware Embedding

Given a bag of sentences* $B^k = \{s_1^k, \ldots, s_{m^k}^k\}$ with the same entity pair (i.e., head entity e_h^k, and tail entity e_t^k) among these sentences, the target of relation extraction is to distinguish the relationship y^k between two given entities. For a clear demonstration, we omit indices of example and sentence in the remainder if no confusion is caused. Each

*"sentence" and "instance" are interchangeable in this section.

sentence is composed of a sequence of tokens, represented as $s = [w_1, \ldots, w_n]$, where n signifies the length of the sentence. Furthermore, each token is represented by a low-dimensional dense vector representation, symbolized as $[\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n] \in \mathbb{R}^{d_w \times n}$, where d_w denotes the dimension of word embedding.

In addition to the typical word embedding, relative position is a crucial feature for relation extraction, which can provide the downstream neural model with rich positional information [1, 21]. Relative position explicitly describes the relative distance between each word w_i and the two targeted entities e_h and e_t. For i-th word, a randomly initialized weight matrix projects the relative position features into two dense-vector representations of the head and tail, i.e., $\boldsymbol{r}_i^{e_h}$ and $\boldsymbol{r}_i^{e_t} \in \mathbb{R}^{d_r}$ respectively. The final low-level representations for all tokens are a concatenation of the aforementioned embeddings, i.e., $\boldsymbol{X}^{(p)} = [\boldsymbol{x}_1^{(p)}, \ldots, \boldsymbol{x}_n^{(p)}] \in \mathbb{R}^{d_p \times n}$ in which $\boldsymbol{x}_i^{(p)} = [\boldsymbol{v_i}; \boldsymbol{r}_i^{e_h}; \boldsymbol{r}_i^{e_t}]$ and $d_p = d_w + 2 \times d_r$.

However, aside from the relative position features, we argue that the embeddings of both the head entity e_h and tail entity e_t are also vitally significant for relation extraction task, since the ultimate goal of this task is to predict the relationship between these two entities. This hypothesis is further verified by our quantitative and qualitative analyses in later experiments (Section 3.3.1, 3.3.2 and 3.3.3). The empirical results show that our proposed embedding can outperform the widely-used way in prior works [67].

In particular, we propose a novel entity-aware word embedding approach to enrich the traditional word embeddings with features of the head and tail entities. To this end, a position-wise gate mechanism is naturally leveraged to dynamically select features between relative position embedding and entity embeddings. Formally, the embeddings of head and tail entities are denoted as $\boldsymbol{v}^{(h)}$ and $\boldsymbol{v}^{(t)}$ respectively. The position-wise gating

procedure is formulated as

$$\boldsymbol{\alpha} = \sigma(\lambda \cdot (\boldsymbol{W}^{(g1)}\boldsymbol{X}^{(e)} + \boldsymbol{b}^{(g1)})), \tag{3.1}$$

$$\tilde{\boldsymbol{X}}^{(p)} = \tanh(\boldsymbol{W}^{(g2)}\boldsymbol{X}^{(p)} + \boldsymbol{b}^{(g2)}), \tag{3.2}$$

$$\boldsymbol{X} = \boldsymbol{\alpha} \cdot \boldsymbol{X}^{(e)} + (1 - \boldsymbol{\alpha}) \cdot \tilde{\boldsymbol{X}}^{(p)}, \tag{3.3}$$

$$\text{where, } \boldsymbol{X}^{(e)} = [\boldsymbol{x}_i^{(e)}]_{i=1}^{n},\ \forall \boldsymbol{x}_i^{(e)} = [\boldsymbol{v}_i; \boldsymbol{v}^{(h)}; \boldsymbol{v}^{(t)}], \tag{3.4}$$

in which $\boldsymbol{W}^{(g1)} \in \mathbb{R}^{d_h \times 3d_w}$ and $\boldsymbol{W}^{(g2)} \in \mathbb{R}^{d_h \times d_p}$ are learnable parameters, λ is a hyper-parameter to control smoothness, and $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{d_h \times n}$, which contains the entity-aware embeddings of all tokens from the sentence.

3.2.2 Self-Attention Enhanced Neural Network

Previous works of relation extraction mainly employ a piecewise convolutional neural network (PCNN) [1] to obtain contextual representation of sentences due to its capability of capturing local features, less computation, and light-weight structure. However, some previous works [7] find that CNNs cannot reach state-of-the-art performance on a majority of natural language processing benchmarks due to a lack of measuring long-term dependency, even if stacking multiple modules. This motivates us to enhance the PCNN with another neural module, which is capable of capturing long-term or global dependencies to produce complementary and more powerful sentence representations.

Hence, we employ a self-attention mechanism in our model due to its parallelizable computation and state-of-the-art performance. Unlike existing frameworks that sequentially stack self-attention and CNN layers in a cascade form [65, 68], we arrange these two modules in parallel so they can generate features describing both local and long-term relations for the same input sequence. Since each bag contains several sentences (up to 20), a light-weight network that can process each sentence efficiently is preferable, such as PCNN, which is the most popular module for relation extraction. For this reason, there is only one self-attention layer in our model. This is different from Yu et al. [65] and Wu

et al. [68] who stack both modules many times repeatedly. Our experiments show that two modules arranged in parallel manner consistently outperform stacking architecture even with additional residual connections [62]). The comparative experiments will be elaborated in Section 3.3.1 and 3.3.2.

Piecewise Convolutional Neural Network This section provides a brief introduction to PCNN as a background for further integration with our model, and we refer readers to [1] for more details. Each sentence is divided into three segments w.r.t. the head and tail entities. Compared to the typical 1D-CNN with max-pooling [21], piecewise pooling has the capability to capture the structure information between two entities. Therefore, instead of using word embeddings with relative position features $\boldsymbol{X}^{(p)}$ as the input, we here employ our entity-aware embedding $\boldsymbol{X}$ as described in Section 3.2.1 to enrich the input features. First, 1D-CNN is invoked over the input, which can be formally represented as

$$\boldsymbol{H} = \text{1D-CNN}(\boldsymbol{X}; \boldsymbol{W}^{(c)}, \boldsymbol{b}^{(c)}) \in \mathbb{R}^{d_c \times n}, \tag{3.5}$$

where, $\boldsymbol{W}^{(c)} \in \mathbb{R}^{d_c \times m \times d_h}$ is convolution kernel with window size of m (i.e., m-gram). Then, to obtain sentence-level representation, a piecewise pooling performs over the output sequence, i.e., $\boldsymbol{H}^{(c)} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n]$, which is formulated as

$$\boldsymbol{s} = \tanh([\text{Pool}(\boldsymbol{H}^{(1)}); \text{Pool}(\boldsymbol{H}^{(2)}); \text{Pool}(\boldsymbol{H}^{(3)})]). \tag{3.6}$$

In particular, $\boldsymbol{H}^{(1)}$, $\boldsymbol{H}^{(2)}$ and $\boldsymbol{H}^{(3)}$ are three consecutive parts of $\boldsymbol{H}$, obtained by dividing $\boldsymbol{H}$ concerning the positions of head and tail entities. Consequently, $\boldsymbol{s} \in \mathbb{R}^{3d_c}$ is the resulting sentence vector representation.

Self-Attention Mechanism To maintain the efficiency of the proposed approach, we adopt the recently-promoted self-attention mechanism [66, 69, 70] for compressing a sequence of token representations into a sentence-level vector representation by exploiting global dependency, rather than computation-consuming pairwise ones [7]. It is used to

measure the contribution or importance of each token to the relation extraction task w.r.t. the global dependency. Formally, given the entity-aware embedding $\boldsymbol{X}$, we first calculate attention probabilities by a parameterized compatibility function, i.e.,

$$\boldsymbol{A} = \boldsymbol{W}^{(a2)}\sigma(\boldsymbol{W}^{(a1)}\boldsymbol{X} + \boldsymbol{b}^{(a1)}) + \boldsymbol{b}^{(a2)}, \tag{3.7}$$

$$\boldsymbol{P}^{(A)} = \mathrm{softmax}(\boldsymbol{A}), \tag{3.8}$$

where, $\boldsymbol{W}^{(a1)}, \boldsymbol{W}^{(a2)} \in \mathbb{R}^{d_h \times d_h}$ are learnable parameters, $\mathrm{softmax}(\cdot)$ is invoked over sequence, and $\boldsymbol{P}^{(A)}$ is resulting attention probability matrix. Then, the result of self-attention mechanism can be calculated as

$$\boldsymbol{u} = \sum \boldsymbol{P}^{(A)} \odot \boldsymbol{X}, \tag{3.9}$$

in which, $\sum$ is performed along sequential dimension and $\odot$ stands for element-wise multiplication. And, $\boldsymbol{u} \in \mathbb{R}^{d_h}$ is also a sentence-level vector representation which is a complement to PCNN-resulting one, i.e., $\boldsymbol{s}$ from Eq.(3.6).

3.2.3 Selective Gate

Given a sentence bag $B = [s_1, \ldots, s_m]$ with common entity pair, where m is the number of sentences. As elaborated in Section 3.2.2, we can obtain $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_m]$ and $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m]$ for each sentence in the bag, which are derived from PCNN and self-attention respectively.

Unlike previous works under multi-instance framework that frequently use a selective attention module to aggregate sentence-level representations into bag-level one, we propose a innovative selective gate mechanism to perform the aggregation. The selective gate can mitigate problems existing in distantly supervised relation extraction and achieve a better empirical effectiveness. Specifically, when handling the noisy instance problem, selective attention tries to produce a distribution over all sentence in a bag; but if there is only one sentence in the bag, even the only sentence is wrongly labeled, the selective attention mechanism will be low-effective or even completely useless. Note that almost

80% of bags from popular relation extraction benchmark consist of only one sentence, and many of them suffer from the wrong label problem. In contrast, our proposed gate mechanism is competent to tackle such case by directly and dynamically aligning low gating value to the wrongly labeled instances and thus preventing noise representation being propagated.

Particularly, a two-layer feed-forward network is applied to each $\boldsymbol{u}_j$ to sentence-wisely produce gating value, which is formally denoted as

$$g_j = \sigma(\boldsymbol{W}^{(g1)}\sigma(\boldsymbol{W}^{(g2)}\boldsymbol{u}_j + \boldsymbol{b}^{(g2)}) + \boldsymbol{b}^{(g1)}), \quad \forall j = 1, \ldots, m, \tag{3.10}$$

where, $\boldsymbol{W}^{(g1)} \in \mathbb{R}^{3d_c \times d_h}$, $\boldsymbol{W}^{(g2)} \in \mathbb{R}^{d_h \times d_h}$, $\sigma(\cdot)$ denotes an activation function and $g_j \in (0, 1)$. Then, given the calculated gating value, a mean aggregation performs over sentence embeddings $[\boldsymbol{s}_j]_{j=1}^m$ in the bag, and thus produces bag-level vector representation for further relation classification. This procedure is formalized as

$$\boldsymbol{c} = \frac{1}{m}\sum_{j=1}^{m} g_j \cdot \boldsymbol{s}_j \tag{3.11}$$

Finally, $\boldsymbol{c}$ is fed into a multi-layer perceptron followed with $|C|$-way $\mathrm{softmax}$ function (i.e., an MLP classifier) to judge the relation between head and tail entities, where $|C|$ is the number of distinctive relation categories. Formally,

$$\boldsymbol{p} = \mathrm{softmax}(\mathrm{MLP}(\boldsymbol{b})) \in \mathbb{R}^{|C|}. \tag{3.12}$$

3.2.4 Model Learning

We minimize negative log-likelihood loss plus L_2 regularization penalty to train the model, which is written as

$$L_{NLL} = -\frac{1}{|\mathcal{D}|}\sum\nolimits_{k=1}^{|\mathcal{D}|} \log \boldsymbol{p}^k_{(i=y^k)} + \beta||\theta||_2^2 \tag{3.13}$$

where $\boldsymbol{p}^k$ is the predicted distribution from Eq.(3.12) for the k-th example in dataset $|\mathcal{D}|$ and y^k is its corresponding distant supervision label.

3.3 Experiment

Table 3.2 : Precision values for the top-100, -200 and -300 relation instances that are randomly selected in terms of one/two/all sentence(s).

Approach	**One**				**Two**				**All**			
P@N (%)	100	200	300	Mean	100	200	300	Mean	100	200	300	Mean
Comparative Approaches												
CNN+ATT [37]	72.0	67.0	59.5	66.2	75.5	69.0	63.3	69.3	74.3	71.5	64.5	70.1
PCNN+ATT [37]	73.3	69.2	60.8	67.8	77.2	71.6	66.1	71.6	76.2	73.1	67.4	72.2
PCNN+ATT+SL [71]	84.0	75.5	68.3	75.9	86.0	77.0	73.3	78.8	87.0	84.5	77.0	82.8
PCNN+HATT [4]	84.0	76.0	69.7	76.6	85.0	76.0	72.7	77.9	88.0	79.5	75.3	80.9
PCNN+BAG-ATT [11]	86.8	77.6	73.9	79.4	91.2	79.2	75.4	81.9	91.8	84.0	78.7	84.8
SeG (*ours*)	**94.0**	**89.0**	**85.0**	**89.3**	**91.0**	**89.0**	**87.0**	**89.0**	**93.0**	**90.0**	**86.0**	**89.3**
Ablations												
SeG w/o Ent	85.0	75.0	67.0	75.6	87.0	79.0	70.0	78.6	85.0	80.0	72.0	79.0
SeG w/o Gate	87.0	85.5	82.7	85.1	89.0	87.0	84.0	86.7	90.0	88.0	85.3	87.7
SeG w/o Gate w/o Self-Attn	86.0	85.0	82.0	84.3	88.0	86.0	83.0	85.7	90.0	86.5	86.0	87.5
SeG w/o ALL	81.0	73.5	67.3	74.0	82.0	75.0	72.3	76.4	81.0	75.0	72.0	76.0
SeG+ATT w/o Gate	89.0	83.5	75.7	82.7	90.0	83.5	77.0	83.5	92.0	82.0	76.7	83.6
SeG+ATT	88.0	81.0	75.0	81.3	87.0	82.5	77.0	82.2	90.0	86.5	81.0	85.8
SeG w/ stack	91.0	88.0	85.0	88.0	91.0	87.0	85.0	87.7	92.0	89.5	86.0	89.1

To evaluate our proposed framework, and to compare the framework with baselines and competitive approaches, we conduct experiments on a popular benchmark dataset for distantly supervised relation extraction. We also conduct an ablation study to separately verify the effectiveness of each proposed component, and last, case study and error analysis are provided for an insight into our model.

Dataset In order to accurately compare the performance of our model, we adopt New York Times (NYT) dataset [35], a widely-used standard benchmark for distantly super-

vised relation extraction in most of previous works [1,4,37,64]. This dataset is generated by aligning Freebase with the New York Times (NYT) corpus automatically. In particular, NYT dataset contains 53 distinct relations including a null class *NA* relation referred to as the relation of an entity pair is unavailable. The training set contains 570K sentences and 293K entity pairs; the test set contains 172K sentences, 96K entity pairs, and 1,950 relational facts (non-NA).

Metrics Following previous works [1, 4, 37, 64], we use precision-recall (PR) curves, Area Under Curve (AUC) and top-N precision (P@N) as metrics in our experiments on the held-out test set from the NYT dataset.

Training Setup For a fair and rational comparison with baselines and competitive approaches, we set most of the hyper-parameters by following prior works [4, 66], and also use 50D word embedding and 5D position embedding released by [4, 37] for initialization, where the dimension of d_h equals to 150. The filter number of CNN d_c equals to 230 and the kernel size m in CNN equals to 3. In output layer, we employ dropout [72] for regularization, where the drop probability is set to 0.5. To minimize the loss function defined in Eq.3.13, we use stochastic gradient descent with initial learning rate of 0.1, and decay the learning rate to one-tenth every 100K steps.

Baselines and Competitive Approaches We compare our proposed approach with extensive previous ones, including feature-engineering, competitive and state-of-the-art approaches, which are briefly summarized in the following.

- **Mintz** [34] is the original distantly supervised approach to solve relation extraction problems with distantly supervised data.
- **MultiR** [36] is a graphical model within a multi-instance learning framework that is able to handle problems with overlapping relations.

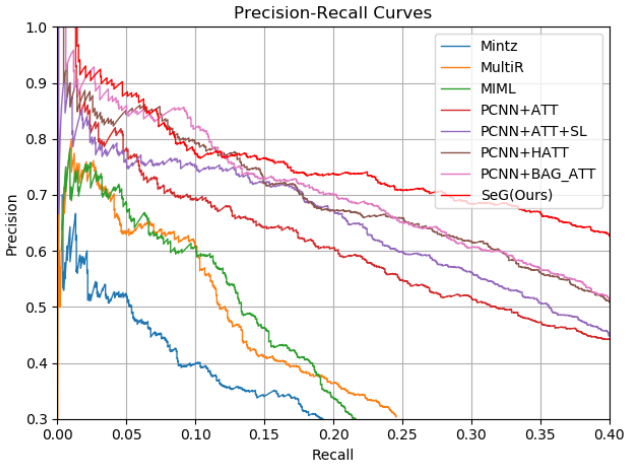

Figure 3.2 : Performance comparison for proposed model and previous baselines in terms of precision-recall curves

- **MIML** [73] is a multi-instance, multi-label learning framework that jointly models both multiple instances and multiple relations.
- **PCNN+ATT** [37] employs a selective attention over multiple instances to alleviate the wrong labeling problem, which is the principal baseline of our work.
- **PCNN+ATT+SL** [71] introduces an entity-pair level denoising method, namely employing a soft label to alleviate the impact of wrong labeling problem.
- **PCNN+HATT** [4] employs hierarchical attention to exploit correlations among relations.
- **PCNN+BAG-ATT** [11] uses an intra-bag to deal with the noise at sentence-level and an inter-bag attention to deal with noise at the bag-level.

3.3.1 Relation Extraction Performance

We first compare our proposed SeG with the aforementioned approaches in Table 3.2 for top-N precision (i.e., P@N). As shown in the top panel of the table, our proposed

Table 3.3 : AUC values of previous work and our model. The comparative results are reported by [4] and [11] respectively.

Approach	**AUC**
PCNN+HATT	0.42
PCNN+ATT-RA+BAG-ATT	0.42
SeG (ours)	0.51

model SeG can consistently and significantly outperform baseline (i.e., PCNN+ATT) and all recently-promoted works in terms of all P@N metrics. In contrast to the end-to-end models, the approaches based on feature engineering perform poorly because an error propagation problem may occur for the pipeline model. Compared to PCNN with selective attention (i.e., PCNN+ATT), our proposed SeG can significantly improve the performance by 23.6% in terms of P@N mean for all sentences; even if a soft label technique is applied (i.e., PCNN+ATT+SL) to alleviate wrong labeling problem in distant supervision, our performance improvement is also very significant, i.e., 7.8%.

Compared to previous state-of-the-art approaches for distantly supervision relation extraction (i.e., PCNN+HATT and PCNN+BAG-ATT), the proposed model can also outperform them by a large margin, i.e., 10.3% and 5.3%, even if they propose sophisticated techniques to handle the noisy training data. These verify the effectiveness of our approach over previous works when solving the wrong or noisy labeling problem that frequently appears in distantly supervised relation extraction.

Moreover, for the proposed approach and comparative ones, we also show AUC curves and available numerical values in Figure 3.2 and Table 3.3 respectively. The empirical results for AUC are coherent with those of P@N, which show that our proposed approach can significantly improve previous ones and reach a new state-of-the-art performance by

handling the wrong labeling problem using a context-aware selective gate mechanism. Specifically, our approach substantially improves both PCNN+HATT and PCNN+BAG-ATT by 21.4% in the aspect of AUC for precision-recall.

3.3.2 Ablation Study

Table 3.4 : AUC values of our model and our model without several components for extensive ablation study.

Approach	**AUC**
SeG (ours)	0.51
SeG w/o Ent	0.40
SeG w/o Gate	0.48
SeG w/o Gate w/o Self-Attn	0.47
SeG w/o ALL	0.40
SeG + ATT w/o Gate	0.47
SeG + ATT	0.47
SeG w/ stack	0.48

To further verify the effectiveness of each module in the proposed framework, we conduct an extensive ablation study in this section. In particular, *SeG w/o Ent* denotes removing entity-aware embedding, *SeG w/o Gate* denotes removing selective gate and concatenating two representations from PCNN and self-attention, *SeG w/o Gate w/o Self-Attn* denotes removing self-attention enhanced selective gate. In addition, we also replace the some parts of the proposed framework with baseline module for an in-depth comparison. *SeG+ATT* denotes replacing mean-pooing with selective attention, and *SeG w/ stack* denotes using stacked PCNN and self-attention rather than in parallel.

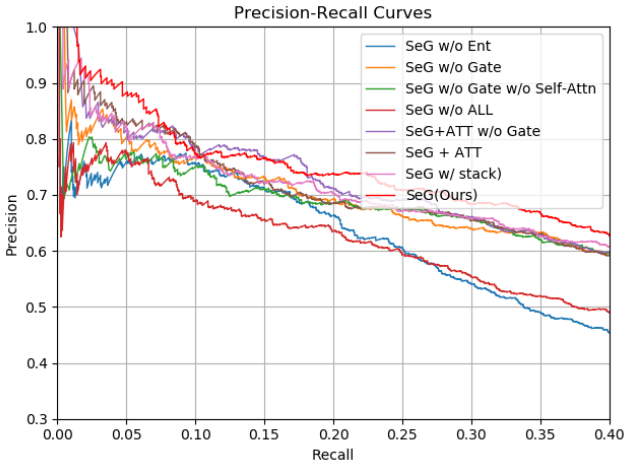

Figure 3.3 : Performance comparison for ablation study under Precision-Recall curves

The P@N results are listed in the bottom panel of Table 3.2, and corresponding AUC results are shown in Table 3.4 and Figure 3.3. According to the results, we find that our proposed modules perform substantially better than those of the baseline in terms of both metrics. Particularly, by removing entity-aware embedding (i.e, SeG w/o Ent) and self-attention enhanced selective gate (i.e., SeG w/o Gate w/o Self-Attn), it shows 11.5% and 1.8% decreases respectively in terms of P@N mean for all sentences. Note that, when dropping both modules above (i.e., SeG w/o ALL), the framework will be degenerated as selective attention baseline [37], which again demonstrates that our proposed framework is superior than the baseline by 15% in terms of P@N mean for all sentences.

Then, we desire to verify whether selective gate module is able to outperform selective attention one when handling wrong labeling problem. Thus, we design the following two setups: 1) we simply replace the selective gate module introduced in Eq.(3.11) with selective attention module, namely, SeG+Attn w/o Gate , and 2) instead of mean pooling in Eq.(3.11), we couple selective gate with selective attention to fulfill aggregation instead mean-pooling, namely, SeG+Attn. Across the board, the proposed SeG still deliver the best results in terms of both metrics even if extra selective attention module is applied.

Table 3.5 : A case study where each bag contains one sentence. *SeG w/o GSA* is an abbreviation of *SeG w/o Gate w/o Self-Attn*.

Bag	Sentence	Relation	SeG (Ours)	SeG w/o Ent	SeG w/o GSA
B1	**Yul Kwon**, 32, of **San Mateo**, Calif., winner of last year's television contest "Survivor" and ...	*/people/person/place_lived*	Correct	Wrong	Wrong
B2	Other winners were Alain Mabanckou from Congo, **Nancy Huston** from **Canada** and Léonora Miano from Cameroon.	*/people/person/nationality*	Correct	Correct	Wrong
B3	... production moved to **Connecticut** to film interiors in places like Stamford, Bridgeport, Shelton, **Ridgefield** and Greenwich.	*/location/location/contains*	Correct	Wrong	Correct
B4	... missionary **George Whitefield**, according to The Encyclopedia of **New York City**.	*NA*	Correct	Wrong	Correct

Lastly, to explore the influence of the way to combine PCNN with self-attention mechanism, as introduced in Section 3.2.2, instead of putting them in parallel, we stack them by following the previous works [65], i.e., SeG w/ Stack. As results derived from our experiments, we observe a notable performance drop after stacking PCNN and self-attention. This verifies our hypothesis that, when a light-weight neural network is relatively shadow, combining self-attention mechanism and PCNN in parallel can achieve a satisfactory result without sacrificing scalability.

3.3.3 Case Study

In this section, we conduct a case study to qualitatively analyze the effects of entity-aware embedding and self-attention enhanced selective gate. The case study of four examples is shown in Table 3.5.

First, comparing Bag 1 and 2, we find that, without the support of the self-attention enhanced selective gate, the model will misclassify both bags into *NA*, leading to a degraded performance. Further, as shown in Bag 2, even if entity-aware embedding module is absent, proposed framework merely depending on selective gate can also make a correct prediction. This finding warrants more investigation into the power of the self-attention enhanced selective gate; hence, the two error cases are shown in Bags 3 and 4.

Then, to further consider the necessity of entity-aware embedding, we show two error cases for SeG w/o Ent whose labels are */location/location/contains* and *NA* respectively in Bag 3 and 4. One possible reason for the misclassification of both cases is that, due to a lack of entity-aware embedding, the remaining position features cannot provide strong information to distinguish complex context with similar relation position pattern w.r.t the two entities.

Chapter 4

Collaborating Relation-Augmented Attention for Hierarchical Ontological Knowledge Learning

4.1 Introduction

Despite being proven to improve overall and long-tail performance, former works also post two issues: 1) Limited by selective attention framework, the relation embeddings are only used as the attention's queries and thus not well-exploited to share knowledge. 2) Despite the capability in mitigating the long-tail problem, graph embeddings pre-trained on large-scale knowledge graphs are time-consuming and not always off-the-shelf, hence at the cost of practicability.

Thus, we propose a novel neural network, named as **Co**llaborating **R**elation-augmented **A**ttention (CoRA), to tackle distantly supervised relation extraction, where no external knowledge is introduced and the relation hierarchies are fully utilized to alleviate the long-tail problem. Specifically, as an alternative to the selective attention framework, we first propose a base model, *relation-augmented attention*, operating at bag level to minimize the effect of wrong labeling, where the relation-augmenting process is fulfilled by sentence-to-relation attention. Empowered by the base model, we then leverage the high-level relations for collaborating features in light of the relation hierarchies. Besides a further relief of wrong labeling, such features facilitate knowledge transfer among the low-level relations inheriting a common high-level relation.

Intuitively, selective attention and its hierarchical extensions learn relation label embeddings to score each sentence in a bag. In contrast, the proposed relation-augmented attention network achieves the same goal via a memory network-like structure: sentences

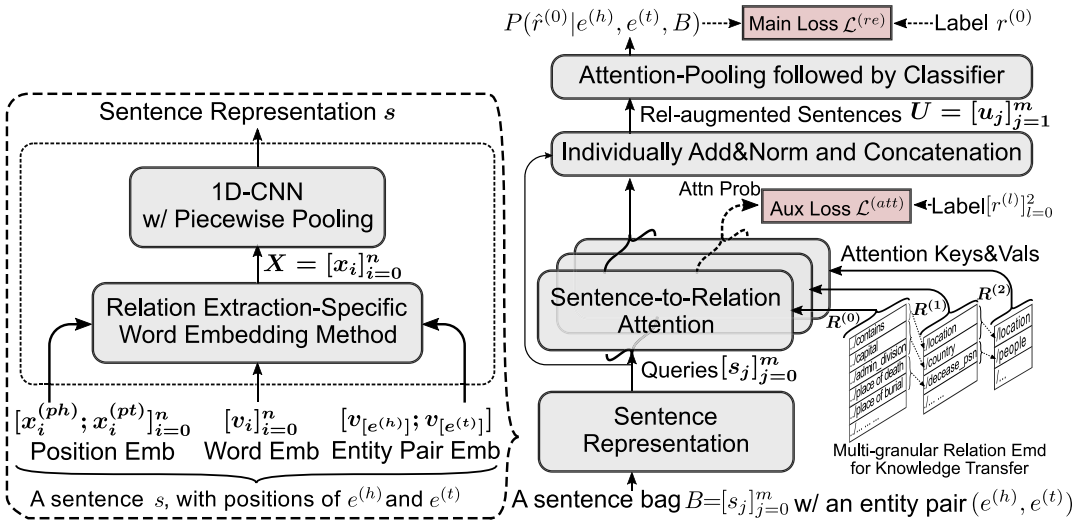

Figure 4.1 : Our proposed **Co**llaborating **R**elation-augmented **A**ttention (CoRA) Network, where the *right part* is the main structure while the *left part* is a sentence embedding method for relation extraction. The illustrated relations and their hierarchies are based on NYT dataset where $M = 2$ in Eq.(4.14).

equipped with relation features are passed into an attention-pooling (i.e., a kind of self-attention [66]) for bag-level representations. Our method is especially effective when extended to multi-granular relations – the features are enriched by cross-relation sharing, which hence benefits long-tail relations.

We use two objectives to jointly train the CoRA. The first is predicting the relation label at bag level, which is the goal of relation extraction. As auxiliary objective, the second is guiding the model to equip each sentence with correct multi-granular relation embeddings during the augmenting process. It aims to boost downstream attention-pooling and is fulfilled by applying the multi-granular labels to the sentence-to-relation attention during training.

4.2 Approach

This section begins with a definition of distantly supervised relation extraction with multi-granular relation labels. Then an embedding method is introduced to represent sentences. Our base model and its hierarchical extension are presented respectively. An illustration of the model is shown in Figure 4.1.

4.2.1 Task Definition

Given a bag of sentences $B = \{s_1, \ldots, s_m\}$ in which each sentence contains a pair of head $e^{(h)}$ and tail $e^{(t)}$ entities in common, the distant supervision [34] assigns this bag with a relation label $r^{(0)}$ according to the entity pair in a knowledge graph. The goal of relation extraction is to predict the relation label $\hat{r}^{(0)}$ of an entity pair based on the corresponding sentence bag when the pair is not included in the knowledge graph. As following the hierarchical setting [4, 5], labels of coarse-grained relations, [$r^{(1)}$, …, $r^{(M)}$], can be used to share knowledge across relations.

4.2.2 Sentence-Level Representation

To embed each sentence s_j in a bag $B = \{s_1, \ldots, s_m\}$ into latent semantic space, we derive a sentence representation from three kinds of features, including word embedding [25], position embedding [1] and entity embedding [74]. The integration of them has been proven crucial and effective to relation extraction by previous work [74]. In the following, we omit the index of a sentence, j, for a clear elaboration. Basically, a sentence s is first tokenized into a sequence of n words, $s = [w_1, \ldots, w_n]$, then a word2vec method [25] is used to transform the discrete tokens into low-dimensional, real-valued vector embeddings, i.e., $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n] \in \mathbb{R}^{d_w \times n}$.

Word Embedding. On the one hand, *position-aware embedding* offers rich positional information for downstream modules [21]. The relative position of the i-th word is repre-

sented by the distances from the word to head $e^{(h)}$ and tail $e^{(t)}$ entities respectively. These two scalars representing the relative distances are then transformed into low-dimensional vectors, $\boldsymbol{x}_i^{(ph)}$ and $\boldsymbol{x}_i^{(pt)} \in \mathbb{R}^{d_p}$, by a learnable weight matrix. Consequently, a sequence of position-aware embeddings is denoted as $\boldsymbol{X}^{(p)} = [\boldsymbol{x}_1^{(p)}, \ldots, \boldsymbol{x}_n^{(p)}] \in \mathbb{R}^{(d_w+2d_p)\times n}$ where $\boldsymbol{x}_i^{(p)} = [\boldsymbol{v}_i, \boldsymbol{x}_i^{(ph)}; \boldsymbol{x}_i^{(pt)}]$. $[;]$ denotes the operation of vector concatenation. On the other hand, *entity-aware embedding* is also crucial since the goal of relation extraction is to discriminate the relation between two entities. The embedding of head or tail entity is represented by the corresponding word embedding. Note that each entity is one entry in the vocabulary of word embedding even if it is usually composed of multiple words. Hence, a sequence of entity-aware embeddings is denoted as $\boldsymbol{X}^{(e)} = [\boldsymbol{x}_1^{(e)}, \ldots, \boldsymbol{x}_n^{(e)}] \in \mathbb{R}^{3d_w\times n}$ where $\boldsymbol{x}^{(e)} = [\boldsymbol{v}_i, \boldsymbol{v}_{[e^{(h)}]}; \boldsymbol{v}_{[e^{(t)}]}] \in \mathbb{R}^{3d_w}$. To integrate the embeddings above, a position-wise gating procedure is employed by following [74]. That is,

$$\boldsymbol{A}^{(e)} = \mathrm{Sigmoid}(\lambda \cdot (\boldsymbol{W}^{(g1)}\boldsymbol{X}^{(e)} + \boldsymbol{b}^{(g1)})), \tag{4.1}$$

$$\tilde{\boldsymbol{X}}^{(p)} = \tanh(\boldsymbol{W}^{(g2)}\boldsymbol{X}^{(p)} + \boldsymbol{b}^{(g2)}), \tag{4.2}$$

$$\boldsymbol{X} = \boldsymbol{A}^{(e)} \circ \boldsymbol{X}^{(e)} + (1 - \boldsymbol{A}^{(e)}) \circ \tilde{\boldsymbol{X}}^{(p)}, \tag{4.3}$$

where "$\circ$" denotes element-wise product $\boldsymbol{W}^{(g1)} \in \mathbb{R}^{d_x\times 3d_w}$ and $\boldsymbol{W}^{(g2)} \in \mathbb{R}^{d_x\times(d_w+2d_p)}$ are learnable parameters, λ is a hyper-parameter to control smoothness, and $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{d_x\times n}$ is the resulting sequence of word embeddings specially for relation extraction.

Piecewise Convolutional Neural Network. As a common practice in distantly supervised relation extraction, piecewise convolutional neural network (PCNN) [1] is used to generate contextualized representations over an input sequence of word embeddings. Compared to the typical 1D-CNN with max-pooling [21], piecewise max-pooling has the capability to capture the structure information between two entities by considering their positions. Specifically, 1D-CNN [55] is first invoked over the input sequence for contextualized representations. Then a piecewise max-pooling performs over the output

sequence to obtain sentence-level embedding. These steps are written as

$$\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n] = \text{1D-CNN}(\boldsymbol{X}; \boldsymbol{W}^{(c)}, \boldsymbol{b}^{(c)}) \in \mathbb{R}^{d_c \times n}, \tag{4.4}$$

$$\boldsymbol{s} = \tanh([\text{Pool}(\boldsymbol{H}^{(1)}); \text{Pool}(\boldsymbol{H}^{(2)}); \text{Pool}(\boldsymbol{H}^{(3)})]), \tag{4.5}$$

where $\boldsymbol{W}^{(c)} \in \mathbb{R}^{d_c \times Q \times d_x}$ is a conv kernel with window size of Q. $\boldsymbol{H}^{(1)}$, $\boldsymbol{H}^{(2)}$ and $\boldsymbol{H}^{(3)}$ are three consecutive parts of $\boldsymbol{H}$, obtained by dividing $\boldsymbol{H}$ w.r.t. indices of head $e^{(h)}$ and tail $e^{(t)}$ entities. Consequently, $\boldsymbol{s} \in \mathbb{R}^{d_h}$, where $d_h = 3d_c$, is the resulting sentence-level representation.

4.2.3 Relation-Augmented Attention Network

Due to the effectiveness of selective attention [37] in multi-instance learning, most recent works employ the selective attention as the baseline and then propose own approaches for improvements in wrong labeling and/or long-tail relations. However, selective attention gradually becomes a bottleneck to performance improvement. For example, [74] find using the simple gating mechanism to replace selective attention further alleviates wrong labeling problem and significantly promotes extracting results. Intuitively, on the one hand, employing the basic PCNN and vanilla attention mechanism inevitably limits the expressive power of this framework and thus sets a barrier. On the other hand, the relation embeddings, similar to label embeddings [75], are crucial to distant supervision relation extraction, but are only used as attention queries to score a sentence and thus not well-exploited.

In contrast, we aim to augment each sentence in a bag with the relation embeddings by sentence-to-relation attention and pass the relation-augmented representations of a bag's sentences into an attention-pooling module. The attention-pooling, a kind of self-attention [6,66,76], is used to derive an accurate bag-level representation for relation classification. In details, we first define a relation embedding matrix $\boldsymbol{R}^{(0)} \in \mathbb{R}^{d_h \times N^{(0)}}$ where d_h denotes the size of hidden states and $N^{(0)}$ denotes the number of distinct relations

$r^{(0)}$ in a distantly supervised relation extraction task. Then, we formulate a sentence-to-relation (sent2rel) attention as opposed to selective attention, which aims at augmenting sentence representation from §4.2.2 with relation information. The sentence representation $\boldsymbol{s}$ is used as a query to attend the relation embedding matrix $\boldsymbol{R}^{(0)}$ via a dot-product compatibility function:

$$\boldsymbol{\alpha}^{(0)} = \mathrm{softmax}(\boldsymbol{s}^T \boldsymbol{R}^{(0)}), \tag{4.6}$$

$$\boldsymbol{c}^{(0)} = \boldsymbol{R}^{(0)} \boldsymbol{\alpha}, \tag{4.7}$$

where $\mathrm{softmax}(\cdot)$ denotes a normalization function along last dimension and $\boldsymbol{c}^{(0)}$ is the resulting relation-aware representation corresponding to the sentence $\boldsymbol{s}$. Then we merge the relation-aware representation $\boldsymbol{c}^{(0)}$ into original sentence representation $\boldsymbol{s}$ by an element-wise gate mechanism with residual connection [62] and layer normalization [63], i.e.,

$$\boldsymbol{\beta}^{(0)} = \mathrm{Sigmoid}(\boldsymbol{W}^{(g)}[\boldsymbol{s}; \boldsymbol{c}^{(0)}] + \boldsymbol{b}^{(g)}), \tag{4.8}$$

$$\tilde{\boldsymbol{u}}^{(0)} = \boldsymbol{\beta}^{(0)} \circ \boldsymbol{s} + (1 - \boldsymbol{\beta}^{(0)}) \circ \boldsymbol{c}^{(0)}, \tag{4.9}$$

$$\boldsymbol{u}^{(0)} = \mathrm{LayerNorm}(\boldsymbol{s} + \mathrm{MLP}(\tilde{\boldsymbol{u}}^{(0)})), \tag{4.10}$$

where $\mathrm{MLP}(\cdot)$ denotes a multi-layer perceptron to increase nonlinearity. Finally, we define relation-augmented sentence representation in our base model as

$$\boldsymbol{u} \coloneqq \boldsymbol{u}^{(0)}. \tag{4.11}$$

Next moving to multi-instance learning, we put each sentence back to its bag $B = \{s_1, \ldots, s_m\}$ so the bag of sentences with relation-augmentation is represented as $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m] \in \mathbb{R}^{d_h \times m}$. Differing from selective attention framework, our sentence representations are augmented by the relation embeddings as elaborated above. Hence, we straightforwardly introduce an attention-pooling module to derive a bag-level representation denoising from the wrongly-labeled sentences. Specifically, the attention-pooling learns to assign each sentence with an importance score according its representation. Then

it performs a weighted sum over a bag of sentence representations, where the weights are proportional to their scores. This attention is formulated as

$$\boldsymbol{b} = \boldsymbol{U}\mathrm{softmax}(\boldsymbol{w}^T\boldsymbol{U}), \tag{4.12}$$

where $\boldsymbol{w}$ is a learnable weight vector, and $\boldsymbol{b}$ denotes the resulting bag-level representation. Lastly, an MLP is used to obtain a categorical distribution over all relations as bag-level prediction:

$$\boldsymbol{p} = P(\hat{r}^{(0)}|e^{(h)}, e^{(t)}, B) \coloneqq \mathrm{MLP}(\boldsymbol{b}) \in \mathbb{R}^{N^{(0)}}. \tag{4.13}$$

4.2.4 Collaborating Relation-Augmented Attention Network

Beyond only fine-grained relations used above, high-level relation embeddings as hierarchical knowledge can collaborate with the low-level embeddings to boost the performance by alleviating long-tail problems [4, 5]. Intuitively, a high-level relation, shared crossing several low-level relations, is used to represent common knowledge of low-level relations. Therefore, via the common high-level relation, 1) several low-level long-tail relations with semantic overlap mutually benefit each other, and 2) the semantic knowledge is easily transferred from data-rich relations to long-tail ones. This common knowledge is implicitly utilized to distinguish the coarse-grained relation of a bag and thus benefits the final relation prediction. With the relation-augmented sentence representation enriched via collaborating, we name it as **Co**llaborating **R**elations-augmented **A**ttention (CoRA).

Empowered by the non-trivial structure design of our base model, high-level relation embeddings can be easily integrated into the base model by re-defining Eq.(4.11). In particular, given the coarse-grained relation labels from low to high level, i.e., $[r^{(1)}, \ldots, r^{(M)}]$, we define a list of relation embedding matrices $[\boldsymbol{R}^{(1)}, \ldots, \boldsymbol{R}^{(M)}]$ in addition to $\boldsymbol{R}^{(0)}$ defined in last section. With these relation embedding matrices, we individually generate their corresponding relation-augmented sentence representations, i.e., $[\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(M)}]$, via the same procedure defined in Eq.(4.6 – 4.10) of §4.2.3. Then, we concatenate

$[\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(M)}]$ in conjunction with $u^{(0)}$ to re-formulate Eq.(4.11) as

$$\boldsymbol{u} \coloneqq [\boldsymbol{u}^{(0)}; \boldsymbol{u}^{(1)}; \ldots, \boldsymbol{u}^{(M)}] \in \mathbb{R}^{(1+M)d_h}. \tag{4.14}$$

The following procedure is identical to that in base model elaborated above, except that the learnable weight matrices are up-scaled linearly with the depth of relation hierarchies.

4.2.5 Training Objectives

The *main objective* for relation extraction is defined to minimize a cross-entropy loss, i.e.,

$$\mathcal{L}^{(re)} = -\frac{1}{|\mathcal{D}|} \sum\nolimits_{B \in \mathcal{D}} \log P(\hat{r}^{(0)} = r^{(0)} | e^{(h)}, e^{(t)}, B), \tag{4.15}$$

where $\mathcal{D}$ is the training set consisting of sentence bags. Besides, an *auxiliary objective* guides sentence-to-relation attention modules to augment each sentence with correct relation embeddings. This is critical to perform downstream attention-pooling and overcome the challenges presented by distant supervision. Given the sent2rel attention score $\boldsymbol{\alpha}^{(l)}$ and relation label $r^{(l)}$ at an arbitrary l level, the loss function to achieve this objective is defined as

$$\mathcal{L}^{(att)} = -\frac{1}{|\mathcal{D}| \cdot |B| \cdot (1+M)} \sum\nolimits_{B \in \mathcal{D}} \sum\nolimits_{s \in B} \sum\nolimits_{l=0}^{M} \log \boldsymbol{\alpha}^{(l)}_{[r^{(l)}]}. \tag{4.16}$$

where $M = 0$ for the base model in §4.2.3, where $M > 0$ for CoRA in §4.2.4. Finally, we optimize the proposed model by jointly minimizing the two loss functions above, i.e., $\mathcal{L} = \mathcal{L}^{(re)} + \mathcal{L}^{(att)}$.

4.3 Experiments

We evaluate our proposed network on a popular benchmark dataset and conduct several analyses for insights into our proposed model.

Table 4.1 : Model Evaluation and ablation study on NYT. "P@N" (top-n precision) denotes precision values for the entity pairs with top-100, -200 and -300 prediction confidences by randomly keeping one/two/all sentence(s) in each bag. *Base model denotes relation-augmented attention network where $M = 0$.

P@N (%)	**One**				**Two**				**All**				**AUC**
	100	200	300	Mean	100	200	300	Mean	100	200	300	Mean	
Comparative Approaches													
CNN+ATT [37]	72.0	67.0	59.5	66.2	75.5	69.0	63.3	69.3	74.3	71.5	64.5	70.1	0.35
PCNN+ATT [37]	73.3	69.2	60.8	67.8	77.2	71.6	66.1	71.6	76.2	73.1	67.4	72.2	0.39
PCNN+HATT [4]	84.0	76.0	69.7	76.6	85.0	76.0	72.7	77.9	88.0	79.5	75.3	80.9	0.42
PCNN+BAG-ATT [11]	86.8	77.6	73.9	79.4	91.2	79.2	75.4	81.9	91.8	84.0	78.7	84.8	0.42
SeG [74]	**94.0**	89.0	**85.0**	**89.3**	91.0	89.0	**87.0**	89.0	93.0	90.0	86.0	89.3	0.51
CoRA (*ours*)	**94.0**	**90.5**	82.0	88.8	**98.0**	**91.0**	86.3	**91.8**	**98.0**	**92.5**	**88.3**	**92.9**	**0.53**
Ablations													
Base* (CoRA w/o Collaborating)	90.0	89.0	85.3	88.1	93.0	90.0	85.3	89.4	93.0	90.5	87.0	90.2	0.52
Base w/o Ent Emb in §4.2.2	83.0	74.0	69.3	74.5	84.0	81.0	72.3	79.1	85.0	80.0	73.3	79.4	0.45
Base w/o Sent2rel Attention in §4.2.3	83.0	74.0	66.6	74.5	82.0	79.0	68.3	76.5	84.0	79.5	73.0	78.8	0.43
Base w/o Attention-pooling in §4.2.3	90.0	87.0	84.0	87.0	93.0	88.0	85.0	88.7	94.0	88.5	86.0	89.5	0.52
Base w/o Aux Obj $\mathcal{L}^{(att)}$ in Eq.(4.16)	80.0	70.0	65.7	71.9	83.0	74.0	68.0	75.0	85.0	80.0	70.3	78.4	0.41

Dataset and Evaluation Metrics. By following previous works [1, 4, 37], we employ the only popular distantly supervised relation extraction dataset, New York Times (NYT) dataset [35]. It contains 53 distinct relations which include a *NA* class denoting the relation between the entity pair is unavailable. And it consists of 570K and 172K sentences in training and test sets respectively. Two metrics, 1) area under precision-recall curve (AUC) and 2) top-n precision (P@N) are usually used to measure the effectiveness. We also use Hits@K for long-tail relations by following [5].

Setups. Following previous works, d_w, d_p, d_x, d_c, d_h and Q are 50, 5, 150, 230, 690 and 3 respectively. λ in Eq.(4.1) is 0.05. NYT offers two more high-level (coarse-grained) relations (i.e., $M = 2$), and the numbers of distinct relations at three levels are 53, 36, and 9. During training, we use minibatch SGD [77] with Adam [78] optimizer. The learning rate is 0.1. The batch size is 160. The dropout probability is set to 0.5. The weight decay of L2 regularization is 10^{-5}.

Comparative Approach. We compare the proposed approach with extensive previous works that are summarized as follows. "$*$" denotes it is proposed for the long-tail problem.

- **PCNN+ATT** [37] proposes a selective attention to alleviate wrong labeling.
- **PCNN+HATT*** [4] employs hierarchical attention to exploit the relations.
- **PCNN+BAG-ATT** [11] proposes intra-bag and inter-bag attentions to handle wrongly-labeled sentences at sentence level and bag level respectively.
- **PCNN+KATT*** [5] integrates externally pre-trained graph embeddings with relation hierarchies for long-tail relations. Note, standard AUC and P@N values are not available while only Hits@K is defined and reported for long-tail settings.
- **SeG** [74] focuses on one-sentence bags and proposes selective gate mechanism.

4.3.1 Evaluation on Benchmark

As shown in Table 4.1 and Figure 4.2 (left), we compare our CoRA with previous competitive approaches on the benchmark in terms of top-n precision, AUC and PR curve. Specifically, CoRA significantly outperforms the selective attention baseline, i.e., PCNN+ATT. It also surpasses the selective gate framework that shows inferior performance on long-tail relations. In addition, compared to PCNN+HATT utilizing relation hierarchies, CoRA achieves much better results in both P@N and AUC.

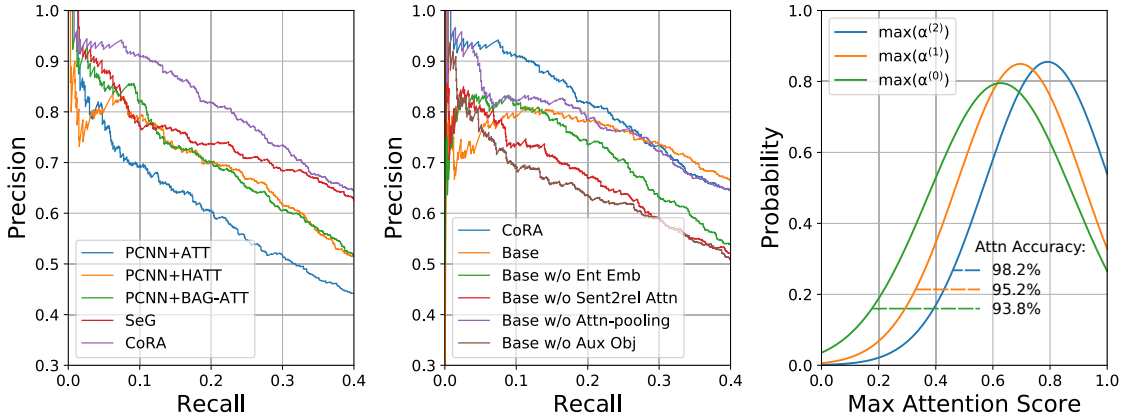

Figure 4.2 : **Left**: Precision-recall (PR) curves on NYT for model comparison. **Middle**: PR curves for ablation study. **Right**: Probability (normal) distribution of maximum attention score, $\max(\boldsymbol{\alpha}^{(l)})$), in sent2rel attention, where attention accuracy is whether the max score $\max(\boldsymbol{\alpha}^{(l)})$ corresponds to $r^{(l)}$.

4.3.2 Ablation Study

To further evaluate the effectiveness of each module in the proposed framework, we conduct an extensive ablation study at the bottom of Table 4.1 and Figure 4.2 (middle). Since the performance drop is consistent in P@N and AUC, we mainly use AUC as the metric to perform the following study. Compared to CoRA, the base model without relation collaborating features only shows a marginal precision drop when the recall > 0.3 in PR-curve, but there is a significant drop on long-tail relations (detailed in the next section). Also, as an alternative to selective attention, our base model outperforms PCNN+ATT by a large margin. Then, removing simple entity embeddings in §4.2.2 leads to remarkable degeneration, verifying its importance. It is also rational to compare PCNN+ATT with "Base w/o Ent Emb" (+0.06 AUC) to demonstrate our relation-augmented framework is indeed better than selective attention. Then, removing "Sent2rel Attention", "Attention-pooling" and "Aux Obj" reduces the AUC by 0.10, 0.01, and 0.12 respectively.

Table 4.2 : Hits@K (Macro) on the relations whose number of training instance $<$ 100/200. "Hits@K" denotes whether a test sentence bag whose gold relation label $r^{(0)}$ falls into top-K relations ranked by their prediction confidences. "Macro" denotes the macro average is applied regarding relation labels.

# Training Instance	**<100**			**<200**		
Hits@K (Macro)	**10**	**15**	**20**	**10**	**15**	**20**
PCNN+ATT [37]	<5.0	7.4	40.7	17.2	24.2	51.5
PCNN+HATT [4]	29.6	51.9	61.1	41.4	60.6	68.2
PCNN+KATT [5]	35.3	62.4	65.1	43.2	61.3	69.2
CoRA	**66.6**	**72.0**	**87.0**	**72.7**	**77.3**	**89.4**
Base	33.3	44.4	66.6	45.5	54.5	72.7
Base w/o Aux Obj	18.5	44.4	61.1	33.3	54.5	68.1
Base w/o Sent2rel Attention	5.0	33.3	61.1	22.7	45.5	68.1

4.3.3 Evaluation on Long-Tail Relations

To prove the capability of CoRA in handling long-tail relations, we conduct an evaluation solely on long-tail relations. Our evaluation setting is identical to [4, 5], where Hits@K (Macro) is used to represent statistical performance on long-tail relations. As shown in Table 4.2, we compare CoRA with competitors and our base models. It is observed that, CoRA improves the performance on long-tail relations by a large margin and delivers a new state-of-the-art results. Compared to previous works (PCNN+HATT/+KATT) that also leverage the relation hierarchies, our relation-augmented attention (Base) without any hierarchy even gets competitive results, not to mention pre-trained graph embeddings used in PCNN+KATT. Further comparing our base model with selective attention (PCNN+ATT), the huge performance gap demonstrates the advantages of our framework

Table 4.3 : Two example sentences with top-3 sent2rel attention scores at all relation levels. Both sentences express the same long-tail relation "*/business/company/founders*".

Example Sentence 1: *Muhammad_yunus*, who won the nobel peace prize, last year, demonstrated with *grameen_bank*, the power of microfinancing.

Top-3 of attention score $\boldsymbol{\alpha}^{(2)}$		Top-3 of attention score $\boldsymbol{\alpha}^{(1)}$		Top-3 of attention score $\boldsymbol{\alpha}^{(0)}$	
/business:	0.422	*NA*	0.383	*NA*	0.387
NA:	0.384	*/business/company*:	0.272	*/business/company/founders*:	0.197
/location:	0.037	*/business/person*:	0.063	*/business/person/company*:	0.063

Example Sentence 2: On sunday, though, there was a significant shift of the tectonic plates of bangladeshi politics, as *muhammad_yunus*, the founder of a microfinance empire, known as the *grameen_bank* and the winner of the 2006 nobel peace prize, announced that he would start a new party and step into the electoral fray.

Top-3 of attention score $\boldsymbol{\alpha}^{(2)}$		Top-3 of attention score $\boldsymbol{\alpha}^{(1)}$		Top-3 of attention score $\boldsymbol{\alpha}^{(0)}$	
/business:	0.755	*/business/company*:	0.679	*/business/company/founders*:	0.652
NA:	0.103	*NA*:	0.089	*NA*:	0.069
/people:	0.031	*/business/person*:	0.059	*/business/person/company*:	0.057

in handling both wrong labeling and long-tail relations. Finally, as shown in the table's last row, removing the proposed sent2rel attention leads to significant decrease, which emphasizes its importance for long-tail relations.

4.3.4 Analysis and Case Study

Distributions of Sent2rel Attention Scores. Sent2rel attention used to incorporate multi-granular relation embeddings is an essential module in CoRA, so its normalized attention scores (i.e., attention probabilities) derived from Eq.(4.6) are critical to measure the knowledge transfer crossing relations. We show a probability distribution of maximum attention score in Figure 4.2 (right). Obviously, a high-level sent2rel attention tends to produce larger maximum attention score and more accurate attention target. It is easily inferred that, 1) accurate attention at high-level promotes the knowledge transfer through the relation hierarchies, and 2) attention probability distribution is more smooth at low-

level to further boost embedding sharing crossing relations. To dig this out, in Table 4.3, we conduct a case study by showing top attention scores at all three relation levels. It is observed that attention scores and the corresponding relations are intuitively consistent with the analyses above. One exception is that *NA* class appears to be assigned with high attention score at low-level sent2rel attention, which indirectly explains 1) our base model w/o collaborating relation features only delivers inferior performance and 2) sent2rel attention for low-level relations are inaccurate.

Performance based solely on Sent2rel Module. Multi-granular relation labels are used as supervision signals for sent2rel attention modules, and the accuracy of each module is greater than 90% as in Figure 4.2. Therefore, it is interesting to check if the attention scores can be directly used to predict relations at the bag level. We present two settings: 1) only using attention scores on fine-grained relations, i.e., $\boldsymbol{\alpha}^{(0)}$, and 2) using products of attention scores at all three levels to make the best of relation hierarchies. Conclusively, the AUC of settings 1 and 2 is 0.41 and 0.43, which outperforms some works in Table 4.1

Error Analysis. To investigate the possible reasons for misclassification, we manually check several randomly-sampled error examples from the test set and find the following factors can cause wrong predictions. 1) Most error cases demonstrate the proposed model still struggles in handling the wrong labeling problem, possibly because the limited expressive power of text representation is incompetent at handling noisy, imbalanced data. 2) The sent2rel attention could be invalid when sibling relations have distinct meanings and post negative effects on relation extraction. For example, */people/person/children* and */people/person/profession* refer to opposite meanings. 3) Since a sentence embedding is augmented by multiple semantically-related relation embeddings, the relation ambiguity problem deteriorates to post errors. For example, it is hard to distinguish */people/deceased_person/place_of_death* and */people/deceased_person/place_of_burial*.

Chapter 5

Hierarchical Relation-Guided Type-Sentence Alignment

5.1 Introduction

To mitigate the long-tail problem, some works [4, 5, 20] resort to the hierarchy of relations for knowledge transfer from data-rich relations to the long-tail ones since the relations have coarse-grained overlap. They focus on interactive operations between hierarchical relations and intra-bag sentences, including relation-to-sentence attention [4] as a hierarchical extension of selective attention, and sentence-to-relation attention [20] enriching sentences with multi-granular relations. As such, they achieve knowledge transfer by learning to distinguish coarse-grained relations for sentences with sufficient data, which provides a latent constraint for the long-tail relations. However, a coarse-grained relation usually denotes the only basic attribute of the distant oracle triple fact in KG, so a sentence scarcely contains its semantics and we can only imply the relation via background information. Again, true-labeled "*Jobs founded Apple*", does not explicitly contain any semantics of its coarse-grained relation "/BUSINESS/COMPANY", but we can directly reason it from the predicate *founded* and type of *Apple*. Thus, it is a challenge for a hierarchical DSRE model to correctly imply coarse-grained relations based solely on sentences, not to mention the existence of the wrong labeling problem.

A direct yet promising way to overcome this challenge is to incorporate extra information for entities [79, 80, 81]. One popular source is the entity types, i.e., an entity's "ISA" attributes in KG, which characterizes the entity from multiple perspectives [82]. As Figure 5.1 shows, although the 1st sentence's semantics is irrelevant to relation, the pairwise types *people.deceased_person* and *location.location* directly align with the fine

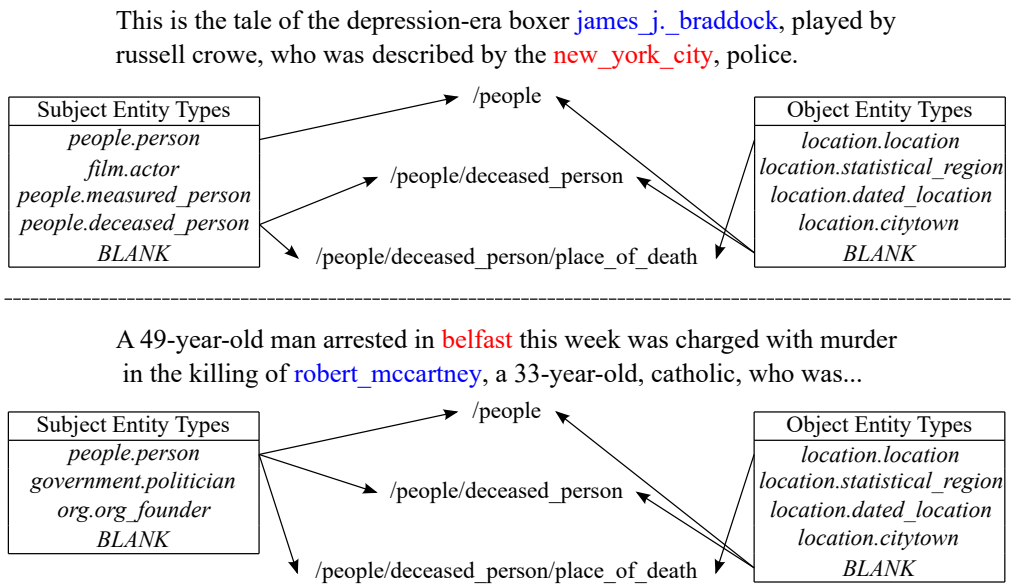

Figure 5.1 : Two sentences with the same long-tail relation. For each sentence, multi-granular relations from top to bottom are pointed by its best pairwise types, which indicates not all pairwise types provide the same contribution. Blue is subject entity, and red is object entity. The 1st sentence relies on the direct pairwise types due to its relation-irrelevant semantics while the 2nd sentence integrates its relation-relevant semantics and pairwise types to enhance its representation.

grained relation. However, existing works [79, 81] ignore this potential of explicit structured types information.

In this work, we aim to improve DSRE by exploiting structured information in the entity types from both pairwise and hierarchical perspectives to alleviate the wrong labeling and the long-tail problems respectively. To this end, we first propose a *context-free type-enriched embedding* module to generate word embeddings with pairwise types associated with the entity pair in a bag. As shown in Figure 5.1, even without the corresponding semantic support, pairwise types can provide direct attributes of entities to align with the relation. Besides, we develop a *context-related type-sentence alignment* module to generate robust sentence representation with pairwise types. Since entities have specific characteristic in certain semantics, we leverage semantics to select proper pairwise types

and then enrich sentence representation, as the 2nd sentence in Figure 5.1 shows. Such an alignment is enhanced by a guidance from the relation to auto-seek for associations between pairwise types and sentences.

At the meantime, hierarchical information has been proven crucial in knowledge transfer for long-tail relations [4, 5, 20]. Thereby, we naturally extend the base alignment module into a hierarchy by proposing a *hierarchical type-sentence alignment* module. An intuitive example in Figure 5.1 shows that different grained relations are pointed by various granular pairwise types. This indicates that these pairwise types contain hierarchical semantics, which makes it feasible to extend base alignment into hierarchy. Thus, the strong association between pairwise types and coarse-grained relations can improve knowledge transfer for long-tail relations.

We conduct extensive experiments on two popular benchmarks, NYT-520k and NYT-570k, showing that our model achieves new state-of-the-art overall and long-tail performance. Further analyses reveal insights into our model.

5.2 Approach

Task Definition. Given a bag of sentences $\mathcal{B} = \{s_1, \ldots, s_N\}$ containing a pair of subject $e^{(s)}$ and object $e^{(o)}$ entities, the distant supervision [34] assigns the sentence bag with a relation label r according to KG triple fact. The goal of relation extraction is to predict the relation label $\hat{r}$ of an entity pair based on the corresponding sentence bag $\mathcal{B}$. Labels of coarse-grained relations, $[r^{(1)}, \ldots, r^{(M)}]$, can be derived from the mention of r. For instance, when $r =$ /BUSINESS/COMPANY/FOUNDERS, $r^{(1)} =$ /BUSINESS/COMPANY and $r^{(2)} =$ /BUSINESS. In the following, we will detail our approach, as illustrated in Figure 5.2.

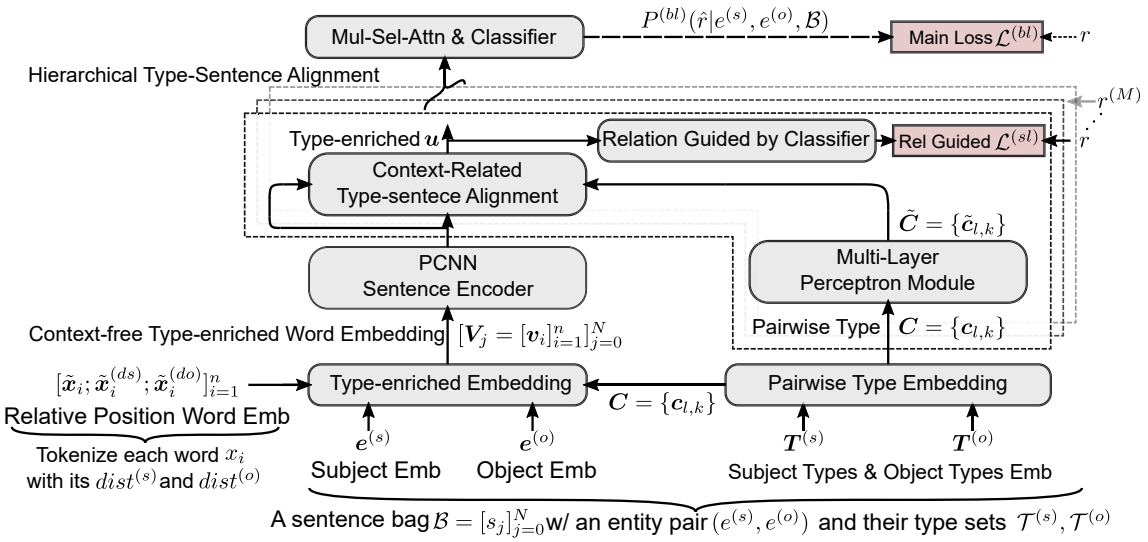

Figure 5.2 : Our proposed model, called **Hi**erarchical **R**elation-guided Type-Sentence **A**lignment **M**odel (HiRAM), for DSRE.

5.2.1 Context-Free Type-Enriched Word Emb

Following most previous DSRE works, we first tokenize each sentence $s_j \in \mathcal{B}$ and employ a word2vec method [25] to derive a sequence of word embeddings by looking up a learnable matrix $\boldsymbol{W}^{(emb)} \in \mathbb{R}^{d_e \times |\mathbb{V}|}$, i.e., $\tilde{\boldsymbol{X}}^j = [\tilde{\boldsymbol{x}}_1^j, \ldots, \tilde{\boldsymbol{x}}_n^j] \in \mathbb{R}^{d_e}$, where $\mathbb{V}$ denotes word vocabulary. j denotes the index of a sentence in the bag and n denotes the sentence length. In the sequel, we omit j if no confusion is caused. Then, as a common practice in DSRE [21], a word's relative distances to both the subject and object entities (a.k.a relative positions) also play significant roles. The distances are first denoted as two integers ($dist^{(s)}$ and $dist^{(o)} \in \mathbb{Z}$) and then embedded into two learnable vectors ($\tilde{\boldsymbol{x}}_i^{(ds)}$ and $\tilde{\boldsymbol{x}}_i^{(do)} \in \mathbb{R}^{d_p}$). Therefore, the updated sequence of word embeddings is $\boldsymbol{X}^j = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$, where $\boldsymbol{x}_i = [\tilde{\boldsymbol{x}}_i; \tilde{\boldsymbol{x}}_i^{(ds)}; \tilde{\boldsymbol{x}}_i^{(do)}] \in \mathbb{R}^{d_w}$, $[;]$ denotes vector concatenation, and $d_w := d_e + 2d_p$.

Previous works [20,74] also found that explicitly enriching each word with both entity embeddings (i.e., $\boldsymbol{e}^{(s)}$ and $\boldsymbol{e}^{(o)}$) in a context-free manner is important to DSRE's success. However, many entities scarcely appear in the raw corpus and have multi-characteristics

(e.g., *Apple* could be a fruit or a company). Thus, the model is hard to distinguish the relations only via sentence semantics. Therefore, we leverage entity types to characterize entities' attributes. That is, given an entity e, its types are defined as a set of type mentions, i.e., $\mathcal{T} = \{t_1, t_2, \dots\}$. However, previous works [81] directly concatenate the entity types of both $e^{(s)}$ and $e^{(o)}$, completely regardless of potentials of explicit structured information of types. As demonstrated by [83], a relation in KG is usually constrained by the entity types of $e^{(s)}$ and $e^{(o)}$ simultaneously (i.e., pairwise types), instead of their individuals. We thereby propose a pairwise type embedding module to enrich the word embedding $\boldsymbol{X}$ also in a context-free manner.

Type and Pairwise Type Embedding. First, given an entity type set $\mathcal{T} = \{t_1, t_2, \dots\}$ (either $\mathcal{T}^{(s)}$ for subject or $\mathcal{T}^{(o)}$ for object), we tokenize each type mention t_j into a sequence of words, then embed the words by looking up $\boldsymbol{W}^{(emb)}$, and lastly derive the type embedding $\boldsymbol{t}_j$ by applying a mean-pooling to the word embeddings of the mention. The embedding of the entire type is

$$\boldsymbol{T} = [\boldsymbol{t}_1, \boldsymbol{t}_2, \dots] \in \mathbb{R}^{|\mathcal{T}| \times d_e}. \tag{5.1}$$

As such, we subsequently define the embedding of the pairwise type by considering a combination of every subject $\forall t_l^{(s)} \in \mathcal{T}^{(s)}$ and object type $\forall t_k^{(o)} \in \mathcal{T}^{(o)}$. Instead of sole semantics via a vector concatenation, we take into account the prior structured information in each type pair by leveraging a translational scheme [56]. Hence, we represent each type pair $(t_l^{(s)}, t_k^{(o)})$ as

$$\begin{aligned} \boldsymbol{c}_{l,k} &= [\tilde{\boldsymbol{c}}_{l,k}^{(sem)}; \tilde{\boldsymbol{c}}_{l,k}^{(str)}] \in \mathbb{R}^{4d_e}, \\ \text{where, } \tilde{\boldsymbol{c}}_{l,k}^{(sem)} &= \boldsymbol{t}_l^{(s)} \odot \boldsymbol{W}^{(sem)} \boldsymbol{t}_k^{(o)}, \\ \text{and } \tilde{\boldsymbol{c}}_{l,k}^{(str)} &= \boldsymbol{t}_k^{(o)} - \boldsymbol{t}_l^{(s)}. \end{aligned} \tag{5.2}$$

Here, "$\odot$" denotes Hadamard product, and $\boldsymbol{W}^{(sem)}$ denotes a learnable projection. $\tilde{\boldsymbol{c}}_{l,k}^{(sem)}$ aims to capture the prior semantic relation in the pair [84] since not all types combinations

are valid in the whole dataset. $\tilde{\boldsymbol{c}}_{l,k}^{(str)}$ aims to measure its structured relation. Lastly, we denote all the embeddings of pairwise types as

$$\boldsymbol{C} = \{\boldsymbol{c}_{l,k}\}_{\forall l \in [1,|\mathcal{T}^{(s)}|], \forall k \in [1,|\mathcal{T}^{(o)}|]}, \tag{5.3}$$

where $\boldsymbol{C} \in \mathbb{R}^{4d_e \times m}$ and $m = |\mathcal{T}^{(s)}| \cdot |\mathcal{T}^{(o)}|$.

Type-Enriched Word Embedding. However, an open question still remains about how to operate on variable-length embeddings of pairwise types, $\boldsymbol{C}$, to enrich each word embedding, $\boldsymbol{x}_j \in \boldsymbol{X}$, in a context-free manner. Inspired by self-attentive sentence encoding [37], we present a bag-level type-attentive module, which compresses $\boldsymbol{C}$ into a single vector representation to facilitate type-enriching. Intuitively, such self-attentive module is focused on the prior knowledge of the type pair in the corpus. Formally, we first generate a global query [37] with structured information of both entities and types to retrieve possible prior pairwise types, i.e.,

$$\tilde{\boldsymbol{q}}^{(f)} = [\boldsymbol{e}^{(o)}; \mathrm{Pool}(\boldsymbol{T}^{(o)})] - [\boldsymbol{e}^{(s)}; \mathrm{Pool}(\boldsymbol{T}^{(s)})], \tag{5.4}$$

followed by a standard Bilinear-based attention,

$$\boldsymbol{q}^{(f)} = \boldsymbol{C} \cdot \mathrm{softmax}(\boldsymbol{C}^T \boldsymbol{W}^{(sa)} \boldsymbol{q}^{(f)}) \in \mathbb{R}^{4d_e}, \tag{5.5}$$

where "$\cdot$" denotes matrix multiplication and $\boldsymbol{W}^{(sa)}$ is a learnable weight matrix. Lastly, we use a gate as in [20] to derive the context-free type-enriched word embedding, i.e.,

$$\boldsymbol{g}_i^{(gf)} = \mathrm{Sigmoid}(\mathrm{MLP}([\boldsymbol{x}_i; \boldsymbol{q}^{(f)}]; \theta^{(gf1)})), \tag{5.6}$$

$$\boldsymbol{x}_i^{(gf)} = \mathrm{MLP}([\boldsymbol{x}_i; \boldsymbol{q}^{(f)}]; \theta^{(gf2)}), \tag{5.7}$$

$$\boldsymbol{v}_i = \boldsymbol{g}_i^{(gf)} \odot \boldsymbol{x}_i + (\boldsymbol{1} - \boldsymbol{g}_i^{(gf)}) \odot \boldsymbol{x}_i^{(gf)}, \tag{5.8}$$

where MLP denotes a multi-layer perceptron (MLP) module. Hence, word embeddings for s are updated to $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n] \in \mathbb{R}^{d_w \times n}$.

5.2.2 Context-Related Type-Sent Alignment

Sentence Encoding. In DSRE, piecewise convolutional neural network (PCNN) [1] is used for sentence embedding. 1D-CNN [55] is first invoked over $\boldsymbol{V}$ for contextualized representations. Then a piecewise max-pooling performs over the output sequence to obtain sentence-level embedding with highlighted entity positions:

$$\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n] = \text{1D-CNN}(\boldsymbol{V}; \theta^{(cnn)}),$$
$$\boldsymbol{s} = \tanh([\text{Pool}(\boldsymbol{H}^{(1)}); \text{Pool}(\boldsymbol{H}^{(2)}); \text{Pool}(\boldsymbol{H}^{(3)})]),$$

where $\boldsymbol{H}^{(1)}$, $\boldsymbol{H}^{(2)}$ and $\boldsymbol{H}^{(3)}$ are three consecutive parts of $\boldsymbol{H}$ by dividing $\boldsymbol{H}$ w.r.t. the indices of subject $e^{(s)}$ and object $e^{(o)}$ entities. Consequently, $\boldsymbol{s} \in \mathbb{R}^{d_h}$ is the resulting sentence-level embedding.

Type-Sentence Alignment. Considering that types are not comprehensive enough to align with multi-granular relations, we leverage semantic context to select valid pairwise types for generating robust sentence representation. Hence, we first calculate alignment scores between a sentence $\boldsymbol{s} \in \mathbb{R}^{d_h}$ and the embeddings of pairwise types $\boldsymbol{C} \in \mathbb{R}^{4d_e \times m}$ by using a simple Bilinear layer, i.e.,

$$\tilde{\boldsymbol{C}} = \text{MLP}(\boldsymbol{C}; \theta^{(p)}) \in \mathbb{R}^{d_h \times m}, \tag{5.9}$$
$$\boldsymbol{a} = \text{softmax}(\tilde{\boldsymbol{C}}^T \boldsymbol{W}^{(al)} \boldsymbol{s}) \in \mathbb{R}^m. \tag{5.10}$$

Then, we enrich the sentence embedding with the aligned type pairs via another gating mechanism:

$$\boldsymbol{z} = \tilde{\boldsymbol{C}} \cdot \boldsymbol{a} \tag{5.11}$$
$$\boldsymbol{g} = \text{Sigmoid}(\text{MLP}([\boldsymbol{s}; \boldsymbol{z}]; \theta^{(g)})), \tag{5.12}$$
$$\tilde{\boldsymbol{u}} = \boldsymbol{g} \odot \boldsymbol{s} + (1 - \boldsymbol{g}) \odot \boldsymbol{z}. \tag{5.13}$$

Lastly, following previous success [8, 20], we leverage a residual connection [62] with layer normalization [63] to derive the final context-related type-enriched sentence embedding, i.e.,

$$\boldsymbol{u} = \mathrm{LayerNorm}(\boldsymbol{s} + \tilde{\boldsymbol{u}}; \theta^{(lm)}). \tag{5.14}$$

Relation-Guided Alignment at the Sentence Level. Due to the severe wrong labeling problem at the sentence level, previous DSRE works usually skip over sentence-level relation supervision. Fortunately, empowered by the proposed context-free type enrichment and context-related type-sentence alignment, we can utilize the sentence-level relation label even if the relation label is wrong. The reason for this is that a sentence has already been equipped with the structured background to support sentence-level relation even if the sentence semantics cannot deliver the relation. We applied an MLP-based neural classifier to the type-enriched sentence embedding $\boldsymbol{u}$, to determine the relation at the sentence level, i.e.,

$$P^{(sl)}(\hat{r}|\boldsymbol{u}) = \mathrm{softmax}(\mathrm{MLP}(\boldsymbol{u}; \theta^{(sl)})), \tag{5.15}$$

where, $P^{(sl)}(\hat{r}|\boldsymbol{u})$ is a categorical distribution over all possible relations. Hence, the training objective is to minimize the cross-entropy loss,

$$\mathcal{L}^{(sl)} = -\sum_{\mathcal{D}}\sum_{\mathcal{B}} \log P^{(sl)}(\hat{r} = r|\boldsymbol{u}), \tag{5.16}$$

where $\mathcal{D}$ denotes a DSRE dataset consisting of sentence bags $\mathcal{B}$. The guidance from the sentence-level relation leads to strong type-sentence alignment (as illustrated in §5.3.1 and §5.3.2). As a result, the sentence-level wrong labeling problem is alleviated. In contrast, previous works w/ sentence-level relation supervisions [85] suffer from the confirmation bias problem [86] caused by the sentence-level wrong labeling.

5.2.3 Hierarchical Type-Sentence Alignment

Inspired by former works [4, 5, 20] for handling long-tail relations, we also extend our basic model into the hierarchy. However, the basic attributes contained by coarse-grained relations are irrelevant to the semantics of sentences. Thus, instead of directly operating on the hierarchy of relations (i.e., from fine-grained r to coarse-grained $[r^{(1)} \dots r^{(M)}]$ relations), we leverage coarse-grained entity types describing the domain/type properties of the entities in the triple facts to enrich each sentence via the guidance from coarse-grained relation.

Formally, we adapt the relation-guided type-sentence alignment (§5.2.2) into the hierarchy, which shares a high-level inspiration with multi-head attention [7]. First, we reuse the architecture from Eq.(5.9-5.14) by defining

$$\begin{aligned} \boldsymbol{a}^{(l)}, \tilde{\boldsymbol{C}}^{(l)} &= \text{TS-Align}^{(l)}(\boldsymbol{s}, \boldsymbol{C}),\ \forall l \in [1, M], \\ \boldsymbol{u}^{(l)} &= \text{TS-Integrate}^{(l)}(\boldsymbol{a}^{(l)}, \tilde{\boldsymbol{C}}^{(l)}, \boldsymbol{s}), \end{aligned} \tag{5.17}$$

where $\text{TS-Align}()$ denotes Eq.(5.9-5.10) to obtain type-sentence alignment $\boldsymbol{a}^{(l)}$ and $\text{TS-Integrate}()$ denotes Eq.(5.11-5.14) to generate enriched sentence representation $\boldsymbol{u}^{(l)}$ at level l. Note that, these modules are parameter-untied from each other. Then, we update the sentence-level relation-guided loss in Eq.(5.16) to its hierarchical version, i.e.,

$$\mathcal{L}^{(sl)} = -\sum_{\mathcal{D},\mathcal{B},l\in[1,M]} \log P^{(sl)}(\hat{r}^{(l)}{=}r^{(l)}|\boldsymbol{u}^{(l)}) \tag{5.18}$$

Again, learnable parameters of the sentence-level classifiers across l are also untied. Lastly, we obtain the hierarchical type-enriched representation, i.e.,

$$\boldsymbol{u}^{(h)} = [\boldsymbol{u}; \boldsymbol{u}^{(1)}; \dots; \boldsymbol{u}^{(M)}] \in \mathbb{R}^{(1+M)d_h}. \tag{5.19}$$

Different from previous works [4, 5, 20] focusing on hierarchical relation embeddings, our work explores the constraints by pairwise types for relations to mitigate sentence-level wrong labeling and uses the hierarchy of entity types on par with that of the relation to improving long-tail performance.

5.2.4 Relation Classification and Objectives

Lastly, we put the sentences back into the bag and derive bag-level embedding for the final relation classification. Hence, for a bag $\mathcal{B} = [s_1, ...s_N]$, we can obtain sentence embeddings of all the sentences $\boldsymbol{U}^{(h)} = [\boldsymbol{u}_1^{(h)}, \ldots, \boldsymbol{u}_N^{(h)}]$, where $\boldsymbol{u}_j^{(h)}$ is hierarchical type-enriched sentence encoding derived from Eq.(5.19). To preserve the hierarchical information learned in $\boldsymbol{u}_j^{(h)}$, we proposed to apply multiple selective modules to its different parts, i.e.,

$$\begin{aligned}
&\boldsymbol{b} = \text{Mul-Sel-Attn}(\boldsymbol{U}^{(h)}) = [\boldsymbol{b}^{(0)}; \boldsymbol{b}^{(1)}; \ldots; \boldsymbol{b}^{(M)}], \\
&\boldsymbol{b}^{(0)} = \text{Selective-Attn}([\boldsymbol{u}_1; \ldots, \boldsymbol{u}_N]), \\
&\boldsymbol{b}^{(l)} = \text{Selective-Attn}([\boldsymbol{u}_1^{(l)}; \ldots, \boldsymbol{u}_N^{(l)}]),\ \forall l \in [1, M].
\end{aligned}$$

where, $\text{Selective-Attn}(\cdot)$ represents the selective attention among the sentences in each granular relation, and $\text{Mul-Sel-Attn}(\cdot)$ represents the selective attention among the multi-granular bag representations. For bag representation, $\boldsymbol{b}^{(0)}$ denotes the finest grained and $\boldsymbol{b}^{(l)}$ denotes coarser grained. Lastly, we use an MLP-based classifier upon $\boldsymbol{b}$ to derive a bag-level categorical distribution, i.e.,

$$P^{(bl)}(\hat{r}|e^{(s)}, e^{(o)}, \mathcal{B}). \tag{5.20}$$

Meanwhile, the corresponding training loss is

$$\mathcal{L}^{(bl)} = -\sum_{\mathcal{D}} P^{(bl)}(\hat{r} = r|e^{(s)}, e^{(o)}, \mathcal{B}). \tag{5.21}$$

Therefore, the final training objective is to minimize a linear combination of both sentence-level in Eq.(5.16) and bag-level (in Eq.(5.21)) losses, i.e.,

$$\mathcal{L} = \mathcal{L}^{(bl)} + \beta \mathcal{L}^{(sl)}. \tag{5.22}$$

5.3 Experiments

Datasets. We evaluate our HiRAM on DSRE benchmarks, New York Times – NYT [35], including NYT-520K and NYT-570K. NYT datasets have 53 distinct relations, in-

cluding an *NA* class denoting the unavailable relation between entity pairs. Each relation includes two coarse-grained relations (i.e., $M = 2$), and the number of relations from fine to coarse are 53, 36 and 9. NYT-520K and NYT-570K have the same testing set containing 172,488 sentences, with 96,678 entity pairs. The only difference is that there is an overlap of 11,416 entity pairs between training and testing in NYT-570K. Thus, NYT-520K has severer wrong labeling and long-tail problems.

Evaluation Metrics. Following previous works [4, 5, 20, 37, 81], we use the area under precision-recall curve (AUC) and top-N precision (P@N) to measure models' performance with the disturbance of wrong labeling and use Hits@K to measure the performance on long-tail relations. AUC measures the ability of relation classification, while P@N measures the precision of high-confidence predictions ranked by the model.

Settings. For both versions of NYT datasets, d_e, d_p, d_w, d_h and M are 50, 5, 60, 690, and 2 respectively. The type number of each entity is various but we set an upper limit and pad BLANK as a choice. We use AdaDelta [77] with 0.1 learning rate. The batch size is 160 with 15 epochs and 5-th is the best, dropout probability is 0.5, weight decay of L2-reg is 10^{-5}. We use random initialization or RoBERTa-base to initialize our models. Whole experiments are computed by a single Titan XP, except for RoBERTa w/ RTX6000.

Comparative Approach. We compare our HiRAM with many strong competitors, including **(1) PCNN+ATT** [37] proposes a selective attention to alleviate wrong labeling. **(2) PCNN+HATT** [4] extends selective attention with hierarchical relations. **(3) RESIDE** [79] leverages side KGs' information to improve DSRE. **(4) PCNN+BAG-ATT** [11] proposes intra-bag and inter-bag attentions to handle the wrongly labeled sentences. **(5) PCNN+KATT** [5] integrates externally pre-trained graph embeddings with relation hierarchies for long-tail relations. **(6) SeG** [74] focuses on one-sentence bags and proposes entity-aware embedding. **(7) CoRA** [20] transfers multi-granular relations

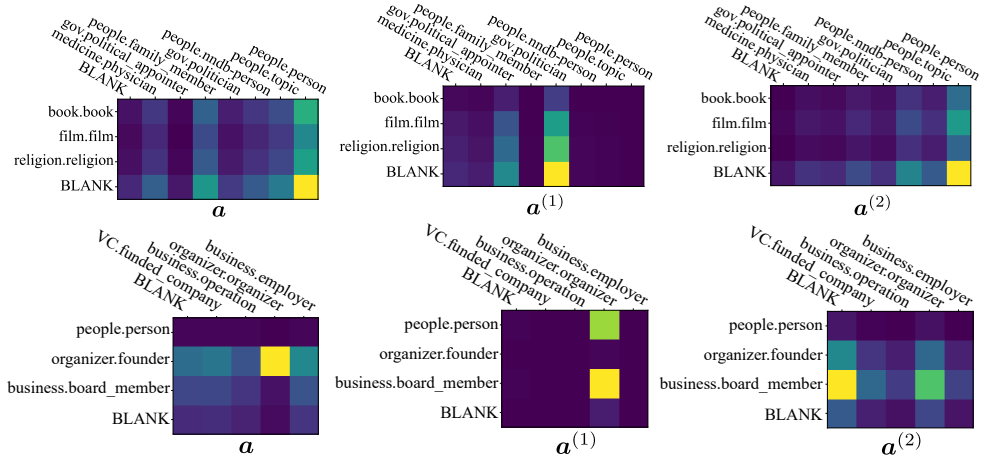

Figure 5.3 : Each heatmap represents the distribution of type-sentence alignment $\boldsymbol{a}$ in Eq.(5.10) and $\boldsymbol{a}^l$ in Eq.(5.17). The horizontal axis represents the types of subject entity, and the vertical axis represents the types of object entity. The top row, from left to right, represents three alignment distributions of first case, and the bottom row represents three alignment distributions of second case, as Table 5.4 shows. Notice that "VC" is the abbreviation of venture captial.

features into sentences in hierarchies for long-tail relations. **(8) InSRL** [81] integrates sentence, entity description and types together via intact space representation learning.

5.3.1 Overall Performance on Benchmarks

As shown in Tables 5.1 and 5.2, HiRAM outperforms former baselines on NYT-570K. Different from CoRA's poor performance on NYT-520K, HiRAM achieves a new state-of-the-art on both popular benchmarks in P@N and AUC. Compared with InSRL integrating both clean entity types' concatenation and accurate entity descriptions, HiRAM increases the AUC score by nearly 7%, verifying the capability of our specific model designer.

5.3.2 Ablation Study

We conduct an ablation study on NYT-520K, as shown at the bottom of Table 5.1. Compared to HiRAM, "HiRAM w/o Hierarchy" drops 6% in AUC. "HiRAM w/o Rel

Guidance" performs well on P@N and AUC but has huge gap in P@One, which represents that the relation-Guided alignment in hierarchy can empower sentence representation with less data in Multi-instance Learning. Meanwhile, top-n precision of "HiRAM w/o CF" drops by nearly 10.5%. To prove the superiority of our specific design, we replace the pairwise type in §5.2.1 with simple type concatenation. The AUC score of "HiRAM w/ TC" decreases by 4.5% and nearly 5.6% of top-n precision. To further emphasize our word embedding §5.2.1 is module-agnostic, we combine RoBERTa [40] with our module respectively. As the bottom panel shows, "RoBERTa w/ CF" makes great progress, and "RoBERTa w/ HiRARM" achieves the best performance among three RoBERTa-related experiments. However, due to the strong ability of RoBERTa model, the wrong labeling problem hurt the performance severely, especially in P@N.

5.3.3 Performance on Long-Tail Relations

Since former baselines are mainly trained on NYT-570K, we reproduce CoRA on NYT-520K for fair comparison as shown in Table 5.3. HiRAM achieves a new state-of-the-art result in Hits@K with 20% superiority. Removing hierarchy module in §5.2.3, the performance of "HiRAM w/o Hierarchy" decreases by nearly 30% on Hits@10 but is better than baselines in other settings, verifying the importance of hierarchical model for long-tail relations. The huge decline of "HiRAM w/o Rel Guidance" verifies the necessity of relation guidance. Due to lacks of plenty reliable training data, RoBERTa is hard to handle the long-tail problem but our specific modules further increase its performance.

5.3.4 Case Study

Firstly, we conduct a case study to qualitatively analyze the effect of our model in §5.2.3 The case study of two samples are shown in Table 5.4 and the type-sentence alignment distribution is shown in Figure 5.3. Secondly, we investigate the possible reasons for the misclassifications of HiRAM.

Distribution of Type-Sentence Alignment. For the first case, despite the failure in expressing the long-tail relation "/PEOPLE/PERSON/RELIGION", the selected pairwise types are sufficient to predict this relation. As the top row of Figure 5.3 shows, *people.person* with *BLANK* helps to identify the character of subject entity, and *religion.religion* with high alignment score can provide direct attributes. For the second case, the semantics is implicitly related to its long-tail relation "/BUSINESS/COMPANY/FOUNDER". The proper pairwise types are selected by coarser relation guidance, like (*organizer.organizer*, *organizer.founder*).

Table 5.1 : Model Evaluation and ablation study on NYT-520K. "P@N" denotes precision values for the entity pairs with the top-100, -200 and -300 prediction confidences by randomly keeping one/two/all sentence(s) in each bag. The abbreviation "CF" represents **C**ontext-**F**ree embedding in §5.2.1; "TC" represents **T**ype **C**oncatenation replacing CF. "RoBERTa" directly predicts relations via [CLS] token. "RoBERTa w/ CF" adds a context-free type-enriched word embedding module on the output of RoBERTa to generate sentence representation. "RoBERTa w/ HiRAM" denotes the combination of HiRAM and RoBERTa.

P@N (%)	**One**				**Two**				**All**				**AUC**
	100	200	300	Mean	100	200	300	Mean	100	200	300	Mean	
Comparative Approaches													
CNN+ATT [37]	76.2	65.2	60.8	67.4	76.2	65.7	62.1	68.0	76.2	68.6	59.8	68.2	-
PCNN+ATT [37]	73.3	69.2	60.8	67.8	77.2	71.6	66.1	71.6	76.2	73.1	67.4	72.2	0.341
CoRA [20]	78.0	69.0	66.0	71.0	79.0	72.0	66.3	72.4	81.0	74.0	68.3	74.4	0.344
RESIDE [79]	80.0	75.5	69.3	74.9	83.0	73,5	70.6	75.7	84.0	78.5	75.6	79.4	-
InSRL [81]	-	-	-	-	-	-	-	-	-	-	-	-	0.451
HiRAM	**93.0**	**89.0**	**83.0**	**88.3**	**93.0**	**88.5**	**84.0**	**88.5**	**93.0**	**88.5**	86.0	**89.2**	**0.484**
Ablations													
HiRAM w/o Hierarchy in §5.2.3	88.0	84.5	83.0	85.2	90.0	86.0	85.0	87.0	90.0	86.5	85.0	87.2	0.450
HiRAM w/o CF in §5.2.1	78.0	75.5	74.3	75.9	87.0	76.5	74.0	79.2	87.0	77.5	74.7	79.7	0.425
HiRAM w/o Rel Guidance in Eq. 5.16	89.0	86.0	76.7	83.9	93.0	88.0	81.7	87.6	93.0	87.0	86.7	88.9	0.482
HiRAM w/ TC	84.0	82.0	75.3	80.4	85.0	81.5	79.7	82.1	89.0	82.5	78.0	83.2	0.462
RoBERTa [40]	44.0	46.5	43.3	44.6	38.0	39.5	38.7	38.7	33.0	36.5	37.7	35.7	0.301
RoBERTa w/ CF	80.0	76.0	74.0	76.7	81.0	78.5	76.0	78.5	81.0	76.0	75.0	77.3	0.488
RoBERTa w/ HiRAM	85.0	83.0	79.3	82.4	86.0	85.5	81.3	84.3	89.0	86.0	81.7	85.6	0.518

Table 5.2 : Model Evaluation on NYT-570K, published by PCNN+HATT [4]

.

P@N (%)	**One**				**Two**				**All**				**AUC**
	100	200	300	Mean	100	200	300	Mean	100	200	300	Mean	
Comparative Approaches													
PCNN+HATT [4]	84.0	76.0	69.7	76.6	85.0	76.0	72.7	77.9	88.0	79.5	75.3	80.9	0.42
PCNN+BAG-ATT [11]	86.8	77.6	73.9	79.4	91.2	79.2	75.4	81.9	91.8	84.0	78.7	84.8	0.42
SeG [74]	94.0	89.0	85.0	89.3	91.0	89.0	87.0	89.0	93.0	90.0	86.0	89.3	0.51
CoRA [20]	94.0	90.5	82.0	88.8	98.0	91.0	86.3	91.8	**98.0**	92.5	88.3	92.9	0.53
HiRAM	**96.0**	**91.5**	**85.7**	**91.1**	**98.0**	**94.5**	**89.3**	**93.9**	**98.0**	**95.0**	**92.3**	**95.8**	**0.580**

Table 5.3 : Hits@K (Macro) tests only on the relations whose number of training instance $<$ 100/200. "Hits@K" denotes whether a test sentence bag whose gold relation label $r^{(0)}$ falls into top-K relations ranked by their prediction confidences."Macro" denotes macro average is applied regarding relation labels. "*" denotes the model is trained on NYT-570K.

# Training Instance	**<100**			**<200**		
Hits@K (Macro)	**10**	**15**	**20**	**10**	**15**	**20**
PCNN+ATT [37]	<5.0	7.4	40.7	17.2	24.2	51.5
PCNN+HATT* [4]	29.6	51.9	61.1	41.4	60.6	68.2
PCNN+KATT* [5]	35.3	62.4	65.1	43.2	61.3	69.2
CoRA* [20]	66.6	72.0	87.0	72.7	77.3	89.4
CoRA [20]	66.6	66.6	75.9	71.7	72.7	80.3
HiRAM	**72.2**	**96.3**	**96.3**	**77.3**	**96.9**	**96.9**
HiRAM w/o Hierarchy in §5.2.3	50.0	88.9	92.6	59.1	90.9	93.9
HiRAM w/o CF in §5.2.1	66.6	88.9	92.6	72.7	90.9	93.9
HiRAM w/o Rel Guidance in Eq. 5.16	55.6	66.7	88.9	63.6	72.7	90.9
HiRAM w/ TC	72.2	77.7	88.9	77.3	81.8	90.9
RoBERTa [40]	0	0	0	0	0	11.6
RoBERTa w/ HiRAM	38.8	61.1	66.6	50.0	54.5	72.7

Table 5.4 : Two cases with long-tail relations are mis-classified by previous works whereas HiRAM is competent. Analysis of the attention probability shown in Figure 5.3 proves the utility of context-related type-sentence alignment with relation guidance.

Case Sentence 1: although the regime of president **bashar_al-assad** hails from an obscure offshoot of shiism – the alawites – syria is nearly three-quarters sunni, with alawites, members of other **muslim** sects and ...		
$r^{(2)}$: /people	$r^{(1)}$: /people/person	$r^{(0)}$: /people/person/religion
Case Sentence 2: having so many operating systems makes it expensive to make software, said **faraz_hoodbhoy**, the chief executive of camera phones save and share multimedia content.		
$r^{(2)}$: /business	$r^{(1)}$: /business/company	$r^{(0)}$: /business/company/founder

Chapter 6

Counterfactual Contrastive Prefix-Tuning

6.1 Introduction

Although fine-tuning paradigm has achieved great success in natural language processing, effectively transferring knowledge to specific tasks, there remains a considerable gap between pre-training and fine-tuning, which can inhibit the transfer and adaptation of knowledge in PLMs to downstream tasks. This gap primarily arises from the diverse objective forms that downstream tasks take on. To narrow this gap, Prompt-tuning [46, 47] has been proposed to unify the objective of different tasks into a cloze-style task to predict target words. Compared to the prevalent fine-tuning, the prompt-tuning paradigm is consistent with language model pre-training and thus generalizable by with few learnable parameters [46, 87, 88, 89].

To bridge the gap to masked language models (MLMs), a task-specific template and verbalizers, are necessary to form a cloze-style task and achieve prompt tuning. Normally, the template can be a natural language prompt or a series of continuous tokens to query the language model, while the verbalizers are usually natural language phrases to represent task-specific labels. For example, in natural language inference (NLI), a training instance can be concatenated with a natural language prompt "[Premise] [MASK] [Hypothesis]". As such, a set of label words is designed as the candidate set for filling into that placeholder (e.g., `[MASK]`) in the designed template. Again, in NLI, the verbalizers are defined as {*Then*, *Maybe* and *But*}, corresponding the three-class categories {*entailment*, *neural* and *contradiction*}. Obviously, it is relatively tractable for experts to select valid label words as there are clearly semantic bounds among these mutual-exclusive labels.

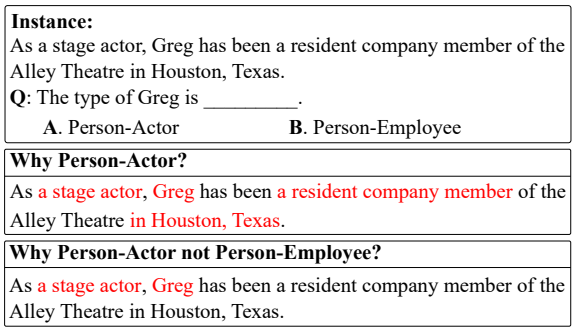

Figure 6.1 : An illustrative example of entity typing task from FewNERD [10] dataset. Option A is its ground-truth label, and Option B is the counterfactual. Red words are the related attributes for the question.

However, with the increase of label space, the semantic boundary among many-class labels becomes obscure, which may overlap leading to the verbalizer ambiguity problem. This explains why some works [90, 91] point out that the performance is quite sensitive to the choice of label words. For instance, as shown in Fig 6.1, "Person-Actor" and "Person-Employee" are the common classes in the entity typing task and share the same hypernym word "Person". To overcome the verbalizer ambiguity problem, [9] manually designs logic rules to merge several sub-prompts together as the final prompt for each class, however, limited by costly expert-required logic rules.

Taking inspiration from the social science research [92], we adopt the contrastive procedure of human explanation to generate diverse information prefixes for training instances. Concretely, rather than explaining "why A", it is more effective to explain "why A not B", where B serves as an implicit counterfactual of A within the current context. In Figure 6.1, we present an instance from the FewNERD [10] dataset, where the task is to classify the type associated with Greg. From a machine learning perspective, a well-trained model will recognize that Greg is associated with multiple attributes, including "Houston", "company" and "actor", all of which are deemed valuable for prediction. As illustrated in Figure 6.1, these contributed attributes can be redundant for prediction as

highlighting. Hence, the contrastive explanation approach tends to overlook most similarity attributes between “Employee” and “Actor”, focusing instead on the more salient semantics that are critical for the model’s differentiation task.

In this chapter, we propose Counter-factual Contrastive Prefix-tuning, dubbed CCPrefix, which aims to minimize semantic obscurity among verbalizers and mitigate the problem of verbalizer ambiguity. Our process begins by constructing all possible fact-counterfactual label pairs, with each class alternately assumed as the fact while the other classes are treated as counterfactuals. Each instance is then projected onto the subspaces spanned by these fact-counterfactual pairs, generating a range of potential contrastive attributes. These potential attributes are subsequently filtered through a global prototype alignment learning method, resulting in an instance-dependent soft prefix. Lastly, we employ a straightforward Siamese representation learning approach for each instance to ensure stability throughout the training process. This methodical multi-step approach strives to reduce ambiguity and enhance the effectiveness of prefix-tuning in the realm of natural language processing.

To comprehensively validate the efficacy of CCPrefix, we conduct extensive experiments on three many-class classification tasks in both fully supervised and few-shot settings, including relation classification, topic classification and entity typing. The experimental results suggest that our work presents a promising step forward in the field, demonstrating the substantial potential of CCPrefidx in handling complex classification tasks in natural language processing.

6.2 Methodology

In this section, we will detail our approach, whose overall architecture is shown in Figure 6.2.

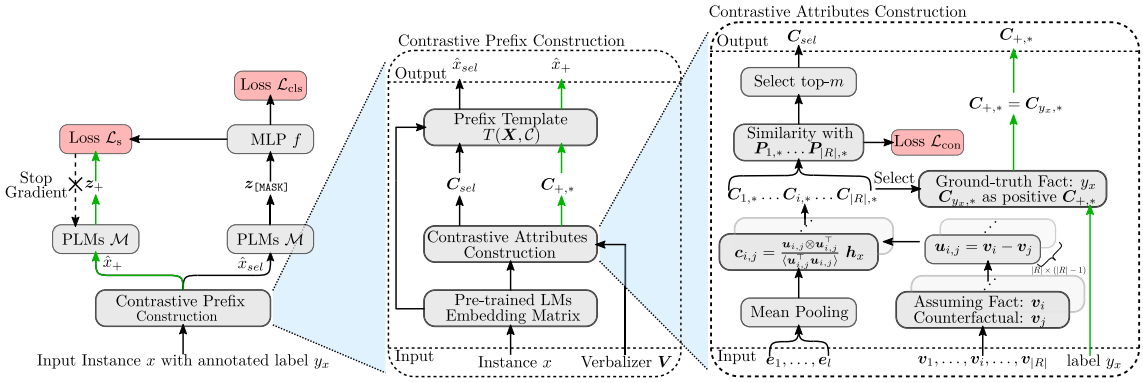

Figure 6.2 : Our proposed model, CCPrefix. For easy comprehension, we zoom in contrastive prefix construction and contrastive attributes generation in Section 6.2.2. The losses $\mathcal{L}_{\text{cls}}$, $\mathcal{L}_{\text{s}}$ and $\mathcal{L}_{\text{con}}$ are defined in Equation (6.9), Equation (6.8) and Equation (6.5). The black line is the forward path for both training and inference, while the green line is the training path with supervised signal.

Task Definition. First of all, we provide the task definition about the classification problem in fine-tuning paradigm. The classification tasks can be denoted as $\mathcal{T} = \{\mathcal{X}, \mathcal{Y}\}$, where $\mathcal{X}$ is the instance set, $\mathcal{Y} = \{y_1, y_2, \dots, y_{|R|}\}$ is the class set, and $|R|$ is the number of classes. The first token of the input is `[CLS]` which contains the special classification embedding. PLMs models take the hidden state $\boldsymbol{h}$ of the first token `[CLS]` as the representation of the whole sequence. A simple softmax classifier is then added to the top of PLMs to predict the probability of class y_c:

$$p(y_c|\boldsymbol{h}) = (\boldsymbol{W}\boldsymbol{h}) \tag{6.1}$$

where $\boldsymbol{W}$ is the task-specific parameter matrix. Both the parameters from PLMs and $\boldsymbol{W}$ will be jointly fine-tuned by maximizing the log-probability of the correct label.

6.2.1 Prefix Tuning for Classification

Formally, prefix tuning consists of a series prefix tokens $\{\boldsymbol{c}_1, \dots, \boldsymbol{c}_m\}$ and a verbalizer $\phi : \mathcal{V} \to \mathcal{Y}$ that bridges the class set $\mathcal{Y}$ and the set of answer words $\mathcal{V}$. To construct the

Algorithm 1 Contrastive Attributes Construction

Input: the class set $\mathcal{Y}$, instance x, a PLM model $\mathcal{M}$

Output: Contrastive attributes $\boldsymbol{C} \in \mathbb{R}^{|R|\times(|R|-1)\times d_e}$

Initialize the verbalizer $\boldsymbol{V} = \phi(\mathcal{Y}) \in \mathbb{R}^{|R|\times d_e}$
Initialize the matrix $\boldsymbol{C} \in \mathbb{R}^{|R|\times(|R|-1)\times d_e}$
Obtain instance representation $\boldsymbol{h}_x = \mathrm{Pool}(\mathcal{M}(x))$
for all $\boldsymbol{v}_i \in \boldsymbol{V}$ **do**
    for all $\boldsymbol{v}_j \in \boldsymbol{V}, i \neq j$ **do**
        Construct the contrastive subspace $\boldsymbol{u}_{i,j} = \boldsymbol{v}_i - \boldsymbol{v}_j \in \mathbb{R}^{d_e}$
        Project the instance onto the subspace $\boldsymbol{c}_{i,j} = \frac{\boldsymbol{u}_{i,j} \otimes \boldsymbol{u}_{i,j}^\top}{\langle \boldsymbol{u}_{i,j}^\top \boldsymbol{u}_{i,j} \rangle} \boldsymbol{h}_x$
    end for
    Form $\boldsymbol{C}_{i,*}$ representing the attributes between i-th fact and the other label
end for
return $\boldsymbol{C} \in \mathbb{R}^{|R|\times(|R|-1)\times d_e}$

cloze-style tasks, at least one placeholder `[MASK]` should be placed into the template for the PLMs, $\mathcal{M}$, as the following shows:

$$T(\boldsymbol{X}, \boldsymbol{C}) = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_l, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_m, \boldsymbol{e}_{\texttt{[MASK]}}\}, \tag{6.2}$$

where $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_l\}$ is the embedding of instance $\boldsymbol{X}$. With the soft prefix template $T(\cdot)$ and the verbalizer ϕ, the learning objective is to maximize $\frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}} \log p(\texttt{[MASK]} = \phi(y_x)|T(x))$.

6.2.2 Contrastive Prefix Construction

We would elaborate on the process of exploring all potential contrastive attributes from each instance and the way we construct the prefex templates.

Contrastive Generation. Thus, for classification tasks, following [93], we construct all causal factors by projecting the sentence representation into the contrastive space.

First of all, each instance x would be encoded by a deep neural encoder $f(\cdot)$ that transforms x into $\boldsymbol{X} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_l\} \in \mathbb{R}^{l \times d_e}$, where l is the sentence length and d_e the embedding dimension. Then, we use a multi-layer perception (MLP) with ReLU activation, and mean pooling over the sequence to get the whole sentence representation, $\boldsymbol{h}_x = \mathrm{Pool}(\mathrm{MLP}(\boldsymbol{X}))$.

Commonly, the prediction of the model $\boldsymbol{W}\boldsymbol{h}_x$ is linear in the latent input representation. The processor of prediction is aim to map $\boldsymbol{h}_x$ to a specific direction $\boldsymbol{w}_i$ via dot product to obtain the logits of class i. As proposed by [93] in terms of contrastive explanation, given two classes, y_p and y_q, if we are particularly interested in the contrastive attributes that the model predicts y_p rather than y_q, we can construct a new basis, $\boldsymbol{u}_{p,q} = \boldsymbol{w}_p - \boldsymbol{w}_q$, which represents a *contrastive space* for y_p and y_q. Thus, y_p is the fact while y_q is one of its counterfactuals. However, for each instance, the golden label is unavailable before prediction. Hence, we hypothesize that the i-th class y_i is the fact in turn while the rest in the finite-label space are counterfactuals to build fact-counterfactual pairs. Specifically, we employ the derivable vectors as the verbalizer $\boldsymbol{V} \in \mathbb{R}^{|R| \times d_e}$ to map to the class set $\mathcal{Y}$. Thus, supposing that i-th class y_i is the fact while one of the rest class y_j is the counterfactual, the contrastive subspace is:

$$\boldsymbol{u}_{i,j} = \boldsymbol{v}_i - \boldsymbol{v}_j \in \mathbb{R}^{d_e}, i \in |R|, j \neq i \tag{6.3}$$

Then, by projecting the instance representation $\boldsymbol{h}_x$ onto the subspace $\boldsymbol{u}_{i,j}$, the contrastive attribute between the specific fact-counterfactual pair is explored:

$$\boldsymbol{c}_{i,j} = \frac{\boldsymbol{u}_{i,j} \otimes \boldsymbol{u}_{i,j}^{\top}}{\langle \boldsymbol{u}_{i,j}^{\top} \boldsymbol{u}_{i,j} \rangle} \boldsymbol{h}_x \tag{6.4}$$

where $\otimes$ is the outer product and $\langle \cdot \rangle$ is the inner product. For the contrastive attributes generated between the same fact and the rest counterfactuals, we denote these attributes as $\boldsymbol{C}_{i,*} \in \mathbb{R}^{(|R|-1) \times d_e}$, where $i, *$ represents the fact-counterfactual pairs consisting of the i-th fact and the rest labels assumed as counterfactuals. Sequentially operating eq.6.3 and

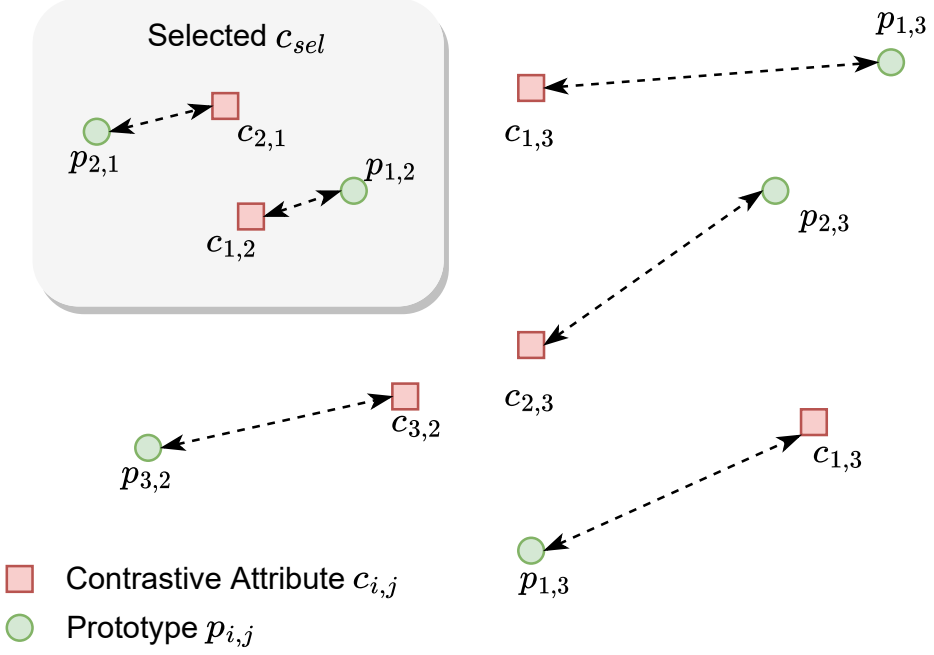

Figure 6.3 : An illustration of the selection process of top-2 contrastive attributes $\boldsymbol{c}_{i,j}$ using the similarities between all possible $\boldsymbol{c}_{i,j}$ and their corresponding prototypes $\boldsymbol{p}_{i,j}$, where i-th class is fact and j-th class is its counterfactual.

eq.6.4, we extract all contrastive attributes $\boldsymbol{C} \in \mathbb{R}^{|R|\times(|R|-1)\times d_e}$ from each instance. We summarize the former procedure of constructing contrastive attributes in Algorithm 1.

Prototype Constraint. Obviously, since we suppose each label as the fact to form fact-counterfactual pairs in turn, it is inevitable to face the noisy attributes projected by invalid fact-counterfactual pairs for each instance. Therefore, the contrastive attributes should be selected only if it is generated by the valid fact-counterfactual pairs formed by the accurate label. To distinguish valid contrastive attributes, we introduce a set of global prototypes $\{\boldsymbol{P}_{0,*}, \boldsymbol{P}_{1,*}, \ldots, \boldsymbol{P}_{|R|,*}\} \in \mathbb{R}^{|R|\times(|R|-1)\times d_e}$ corresponding to contrastive attributes. Concretely, for the contrastive attributes $\boldsymbol{c}_{i,j}$ generated by projecting instance onto the subspace between i-th fact and j-th counterfactual, there is only one corresponding prototype $\boldsymbol{p}_{i,j}$. The fine-grained global prototypes can learn the common features of their

corresponding fact-counterfactual attribute among the whole training instances. During training, according to the instance's ground-truth label, these prototypes can be split into two groups. One is the set of positive prototypes while the other is the rest negative prototypes $\boldsymbol{P}_{-,*} \in \mathbb{R}^{(|R|-1)\times(|R|-1)\times d_e}$. The positive prototypes represent the common knowledge of the corresponding attributes $\boldsymbol{C}_{+,*}$ generated by the valid fact-counterfactual pairs. These prototypes are trained with the following self-contrastive learning loss:

$$\mathcal{L}_{\text{con}} = -\log \frac{\exp(\langle \boldsymbol{W}\boldsymbol{C}_{+,*}, \boldsymbol{P}_{+,*} \rangle)}{\sum_{-} \exp(\langle \boldsymbol{W}\boldsymbol{C}_{+,*}, \boldsymbol{P}_{-,*} \rangle))} \tag{6.5}$$

where $\boldsymbol{W} \in \mathbb{R}^{d_e \times d_e}$ is the learning weight matrix and $\langle \cdot \rangle$ is the inner product to calculate the similarity. This objective forces the positive prototypes to draw up positive contrastive attributes. Simultaneously, the negative contrastive attributes would be pushed away from the positive prototypes.

Prefix Construction. Thus, by calculating the similarities between the instance's contrastive attributes and the corresponding prototypes, we select the top-m's most similar attributes $\boldsymbol{C}_{sel} \in \mathbb{R}^{m \times d_e}$ as additional prefix tokens, as shown in Figure 6.3. The selected contrastive attributes will be considered as a series of tokens in the prefix template $T(\cdot)$, as Equation (6.2).

6.2.3 Siamese Prefix Tuning Objective

We note that some selected top-m contrastive attributes may inevitably take false classes as facts, thereby introducing unwanted noise. Therefore, it is crucial to force the PLMs to focus on the valid contrastive attributes and consequently stabilize the model performance. Hence, we leverage a simple Siamese representation learning method [94] to simultaneously train the PLMs, $\mathcal{M}$, via maximizing the similarity between the prefix templates with selected contrastive attributes $\boldsymbol{C}_{sel}$ and the same instance with all positive attributes $\boldsymbol{C}_{+,*}$. These two inputs with different contrastive attributes are fed into $\mathcal{M}$ to

obtain the [MASK] representation $\boldsymbol{z}$ and $\boldsymbol{z}_+$:

$$\begin{aligned} \boldsymbol{z} &= \mathcal{M}(\hat{\boldsymbol{X}}) = T(\boldsymbol{X}, \boldsymbol{C}_{sel}), \\ \boldsymbol{z}_+ &= \mathcal{M}(\hat{\boldsymbol{X}}_+) = T(\boldsymbol{X}, \boldsymbol{C}_{+,*}). \end{aligned} \tag{6.6}$$

Then, we minimize the negative cosine similarity between two outputs with an MLP $f(\cdot)$:

$$\mathcal{D}(\boldsymbol{z}, \boldsymbol{z}_+) = -\frac{f(\boldsymbol{z})}{||f(\boldsymbol{z})||_2} \cdot \frac{\boldsymbol{z}_+}{||\boldsymbol{z}_+||_2} \tag{6.7}$$

Following [94], we use a symmetrized loss with the stop-gradient operation:

$$\mathcal{L}_{\mathrm{s}} = \frac{1}{2}\mathcal{D}(f(\boldsymbol{z}), (\boldsymbol{z}_+)) + \frac{1}{2}\mathcal{D}(f(\boldsymbol{z}_+), (\boldsymbol{z})). \tag{6.8}$$

Here, $\boldsymbol{X}$ with attributes $\boldsymbol{C}_{+,*}$ receives no gradient from $\boldsymbol{z}_+$ in the first term, but it receives gradients from $f(\boldsymbol{z}_+)$ in the second term, and vice versa.

Finally, the learning objective is to minimize the following loss:

$$\mathcal{L}_{\mathrm{cls}} = -\frac{1}{|\mathcal{X}|}\sum_{k=1}^{|\mathcal{X}|} \log p(\texttt{[MASK]} = \boldsymbol{v}_k | x_k) \tag{6.9}$$

where $p(\texttt{[MASK]} = v_k | x_k)$ is the predicted distribution for the k-th sample in dataset $\mathcal{X}$ and $\boldsymbol{v}_k$ is the answer word corresponding to its ground truth label y_k. Overall, our final training loss is

$$\mathcal{L} = \mathcal{L}_{\mathrm{cls}} + \mathcal{L}_{\mathrm{s}} + \mathcal{L}_{\mathrm{con}} \tag{6.10}$$

6.3 Experiments

We conduct experiments on several classification tasks, including relation classification (RC), topic classification (TC) and entity typing (ET).

6.3.1 Datasets

We adopt 4 popular datasets for relation classification, i.e., TACRED [95], TACREV [96], ReTACRED [97] and SemEval 2010 Task 8 [98] (SemEval), one for topic classification, i.e., DBPedia [99], and one for entity typing, i.e., FewNERD [10].

- **TACRED**, **TACREV** and **ReTACRED** are used widely for relation classification. While TACRED is the origin, TACREV and ReTACRED are its revised versions with modifications in test sets and some relation tpyes.
- **SemEval** is a traditional dataset for RC.
- **DBPedia** is an ontology dataset with structured information extracted from WikiPedia. We privately set a 10% of the training dataset as the validation set.
- **FewNERD** is a manually large-scale dataset of entity typing containing 66 fine-grained entity types. We focus on the inter-task, where train/dev/test splits may share coarse-grained types while keeping the fine-grained entity types mutually disjoint.

More details of these datasets are shown in Table 6.1. For evaluation, we use F_1 scores as the metric for RC, and mean accuracy for TC and ET.

Dataset	#Class	Task	$\lvert\mathcal{D}_{\text{train}}\rvert$	$\lvert\mathcal{D}_{\text{dev}}\rvert$	$\lvert\mathcal{D}_{\text{test}}\rvert$
TACRED	42	RC	68,124	22,631	15,509
TACREV	42	RC	68,124	22,631	15,509
ReTACRED	40	RC	58,465	19,584	13,418
SemEval	19	RC	6,507	1,493	2,717
DBPedia	14	TC	56,000	5,600	70,000
FewNERD	66	ET	338,753	48,667	96,901

Table 6.1 : Basic statistics of the datasets, where RC stands for relation classification, TC stands for topic classification, and ET stands for entity typing.

	Extra Data	TACRED	TACREV	ReTACRED	SemEval
C-GCN [100]	-	66.3	74.6	80.3	-
RoBERTa$_{\text{LARGE}}$ [40]	-	68.7	76.0	84.9	87.6
KnowBERT [101]	✓	71.5	79.3	-	89.1
SpanBERT [102]	✓	70.8	78.0	85.3	-
LUKE [103]	✓	**72.7**	80.6	90.3	-
PTR [9]	-	72.4	81.4	90.9	89.9
CCPrefix (Ours)	-	72.6	**82.9**	**91.2**	**90.6**
w/o ConAtt in §6.2.2	-	70.0	80.9	90.6	90.1
w/o Prototypes in §6.2.2	-	71.9	81.2	90.5	90.4
w/o $\mathcal{L}_{\text{con}}$ in Eq.6.5	-	71.3	81.8	90.6	90.2
w/o Siamese in §6.2.3	-	72.0	81.8	90.8	90.1

Table 6.2 : F_1 scores (%) for RC tasks on the 4 datasets in the fully supervised setting. "w/o ConAtt" denotes using manually Prefix template and soft verbalizer. "w/o Prototypes" denotes that the cluster is rely on the verbalizer. "w/o Siamese" denotes that the input of Prefixs template only maintain instance and selected contrastive attribute.

6.3.2 Settings

To fairly compare with SoTA baselines, we evaluate CCPrefix under fully supervised and few-shot settings for RC tasks, and exclusively in few-shot settings for TC and ET, where for each class, K instances are sampled for training and validation. Following previous works [9, 104], we set K as 8, 16, 32 for relation classification and 1, 2, 4, 8, 16 for topic classification and entity typing. We use a fixed set of 5 random seeds to sample instances and take the average of all results as the final result.

	TACRED			TACREV			ReTACRED		
	8	16	32	8	16	32	8	16	32
Fine-Tuning (Ours)	12.2	21.5	28.0	13.5	22.3	28.2	28.5	49.5	56.0
PTR [9]	28.1	30.7	32.1	28.7	31.4	32.4	51.5	56.2	62.1
CCPrefix (Ours)	**30.1**	**33.4**	**37.6**	**29.8**	**33.0**	**34.0**	**54.5**	**61.4**	**65.2**
w/o ConAtt in §6.2.2	18.1	29.6	32.6	18.1	29.0	32.7	41.1	55.5	64.1
w/o Prototypes in §6.2.2	28.5	33.1	36.3	30.4	31.7	33.2	54.2	56.3	62.1
w/o $\mathcal{L}_{\text{con}}$ in Eq.6.5	28.2	33.2	37.3	28.9	32.1	33.8	53.5	59.7	64.4
w/o Siamese in §6.2.3	23.8	33.1	32.9	27.9	30.4	33.2	50.6	57.7	63.4

Table 6.3 : F_1 scores (%) for RC tasks in the few-shot setting. We use $K = 8, 16, 32$ for few-shot settings.

6.3.3 Implementation Details

Our model is implemented based on PyTorch [105] with V100 and the Transformer repository of Huggingface [106]. For RC and TC tasks, our model is based on ROBERTA$_{\text{LARGE}}$ [40], while for ET, it is based on BERT$_{\text{BASE}}$ [8]. Adam optimizer [78] is used for all datasets, where the learning rate is manually tuned $\in$ {$1e$-5, $3e$-5, $5e$-5 }, and the decay rate is set to $1e$-2, and the batch size is set to 16. For the fully-supervised setting, the epoch is 5 while for few-shot setting, it is 30. The best model is selected based on the performance on the development set. We select top-m attributes as prefix, where $m = |R| - 1$.

6.3.4 Comparison Methods

We mainly compare CCPrefix with several representative methods in many-class classification tasks, including learning-from-scratch methods, fine-tuning methods and Prefix-

tuning methods. 1) C-GCN [100] is a learning-from-scratch based on graph neural networks for relation classification. 2) For fine-tuning vanilla PLMs, we directly select ROBERTA$_{\text{LARGE}}$ as our baselines for relation classification. 3) Since entity information is crucial in relation classification, we select SPANBERT [102], KNOWBERT [101] and LUKE [103] as our baselines. 4) We select PTR [9], a prompt augmentation model, for relation classification. 5) For topic classification and entity typing, our baselines are ProtoVerb [104] that uses manual prompts, and PETAL [47] that extracts words as prompts.

	DBPedia					FewNERD				
	1	2	4	8	16	1	2	4	8	16
PETAL [47]	60.06	78.21	86.40	88.41	92.90	20.88	31.28	43.10	50.78	55.49
ProtoVerb [104]	72.85	85.49	90.91	95.75	96.30	**25.00**	**35.72**	48.28	56.06	61.29
CCPrefix (Ours)	**84.02**	**93.26**	**95.17**	**97.66**	**98.45**	22.78	32.47	**51.49**	**58.54**	**63.38**

Table 6.4 : Few-Shot TC & ET performance of F_1 scores (%) on the DBPedia and FewNERD datasets. We use $K = 1, 2, 4, 8, 16$ for few-shot settings.

6.3.5 Main Quantitative Evaluation

We compare CCPrefix with several recent methods to conduct an in-depth analysis.

Fully Supervised Setting As indicated in Table 6.2, CCPrefix significantly outperforms former baselines, even surpassing KNOWBERT and LUKE that leverage external task-specific knowledge to enhance models. Compared to PTR [9], which manually constructs logic rules as the prompt, CCPrefix even outperforms. Such comparison indicates that the unique task-related information to form a unique prefix can better stimulate task-specific knowledge in PLMs.

Few-Shot Setting To further assess our model, we evaluate CCPrefix in few-shot settings. For relation classification, as shown in Table 6.3, CCPrefix outperforms PTR, with an average improvement of 6.6% on ReTACRED. For topic classification, as shown in the left panel of Table 6.4, CCPrefix exceeds PETAL and ProtoVerb by a large margin. Specifically, in the extreme data scarce scenario ($K = 1, 2$), our model surpasses ProtoVerb by 15.3% and 9.1%. This demonstrates that, if the class labels are semantically diverse, our model is capable of acquiring sufficient knowledge from the PLM even in this limit. For entity typing, our model exceeds former baseline in several scenarios ($K = 4, 8, 16$) but not good when training instances are extremely scarce ($K = 1, 2$). We infer that for fine-grained entity typing, although our model can cancel out most of the attributes between two classes sharing the same coarse class with subtle differences in semantic (e.g., 'building-theater" and "building-library" are under type "building"), it is hard to discriminate such contrastive attributes in extreme data scarce scenario.

6.3.6 Ablation Study

We carry out an ablation study on relation classification datasets to further invetigate the effectiveness of each component in CCPrefix, as detailed in the bottom panel of Table 6.2 and Table 6.3. "w/o ConAtt" causes more performance degradation in the few-shot setting than in the fully supervised one, which indicates that contrastive attributes can further stimulate the knowledge in PLMs. For "w/o Prototypes", attribute-verbalizer similarities are used as the slection criteria, causing a significant performance drop due to noise attributes, although it slightly outperforms CCPrefix in TACREV under K=8. "w/o $\mathcal{L}_{\text{con}}$" has less performance reduction in few-shot setting than that in fully supervised setting. We infer that the unbalanced training data distribution may hurt the performance significantly. The performance of "w/o Siamese" drops severely in the extreme data scarce scenario ($K = 8$), indicating that simple representation learning can force the PLMs to focus on the valid contrastive attributes in prefix.

Relation	Top selected counterfact
per:siblings	*per:title*
per:parents	*per:countries_of_residence*
org:dissolved	*org:member_of*
per:origin	*org:dissolved*
per:children	*per:country_of_birth*
per:city_of_birth	*per:city_of_death*
per:employee_of	*per:countries_of_residence*
per:religion	*per:city_of_death*
org:alternate_names	*org:founded_by*
per:cause_of_death	*per:country_of_death*
org:website	*org:members*

Table 6.5 : The top selected counterfactual relation learned by the model for some relation types.

6.3.7 Selected Counterfact

Since the prefixes are instance aware, we limit our analysis to a subset of 7K instances in the test set that could be correctly classified. For each relation type, we count the most frequently selected counterfactual relation. Part of the results are shown in Table 6.5. It is notable that most of the time the model can match a pair *per* relations, or a pair of *org* relations. Also, the model prefers to select two relation types semantically correlated but with subtle differences. For example, for relation *per:city_of_birth* or *org:dissolved*, the corresponding contrastive attribute factor is *per:city_of_death* or *org:member_of*.

6.3.8 Case Study

To analyze the influence of individual tokens on model prediction, we conduct a case study on the relation *per:city_of_birth* between entities "`he`" and "`Potomac`". "`Potomac`",

y^*=*per:city_of_birth*	(y^*, *y'*)=*per:city_of_birth, per:city_of_death*
Gross , a 60-year-old native of Potomac , Maryland , was working for a firm contracted by USAID when he was arrested Dec 3 , 2009 , and sent to Cuba 's high-security Villa Marista prison .	Gross , a 60-year-old native of Potomac , Maryland , was working for a firm contracted by USAID when he was arrested Dec 3 , 2009 , and sent to Cuba 's high-security Villa Marista prison .

Figure 6.4 : The highlighted tokens of the same sentence where the two entities are underscored. On the left, the tokens are projected onto the ground truth y^*=*per:city_of_birth*, and on the right onto the contrastive space between y^* and the counterfactual *y'*=*per:city_of_death*.

as depicted in Figure 6.4. We compute the similarity between each word and the fact y^*=*per:city_of_birth*, as well as the contrastive attribution factor between y^*=*per:city_of_birth* and *y'*=*per:city_of_death*. For clarity, words with similarity scores exceeding the average are highlighted. Our results reveal that the contrastive attribute factor yields concentrated, key determinant highlights such as "native of". In contrast, using y^* alone results in scattered highlights, diverging from human expectations of the significant predictors.

6.3.9 Error Analysis

Our model operates under the strong assumption that all labels, save for the golden one, act as counterfactuals of the golden label. This hypothesis neglects the semantic correlations and overlaps among different classes, potentially impacting model performance. This issue is especially apparent in the entity typing task, where fine-grained entity types

mayu semantically overlap, thereby challenging our assumption. When class labels possess subtly distinct semantics, more data is needed to construct valid contrastive attributes. This can cause model performance to drop in scenarios of extreme data scarcity, like with the FewNED dataset at $K = 1, 2$.

6.4 Related Work

Prefix Tuning in Classification. The templates can be categorized into two groups, i.e., discrete prompt [46, 47, 49] and continuous prefix [53, 54]. Discrete prompts often manually designed for all training instances with task descriptions. [9] leverage manual logic rules to combine label-related sub-prompts together. Although it is a concrete manifestation of human's interpretation of the task, discrete prompts may not be the optimal solution. Continuous prefixes [53,54], attached to instances, have proven useful but fail to fully capture the diversity of training instances. Our work inspired by the human decision process, introduces an instance-dependent prefix, better addressing the discrimination of label space.

Verbalier in Classification. Reformulating problems as language modeling tasks has been explored in few-shot scenarios [46, 87, 88, 89]. Traditional manual verbalizer mappings demand expert knowledge, thus making automatic verbalizer search [47, 49] an appealing alternative. This approach iteratively enhances the label-to-word mapping in a greedy fashion.

Counterfactual Contrastive. Explanation of artificial intelligence is widely concerned in recent years. [92] presents the philosophical foundations of explanation that human relies on the contrastive explanations. [93] highlights the attributes in the latent space to provide fine-grained explanation of model decision. Furthermore, [107] produces contrastive explanations by editing the inputs for the contrast case while [108] uses it for

evaluation. [109] builds contrastive prompts with instance-specific information for explanation. [110] employs contrastive counterfactuals with the multi-instance framework for vision-language grounding. [111] tasks humans with revising dataset to revise the dataset with counterfactuals. Meanwhile, [112] produces high-quality augmented data with counterfactuals to overcome out-of-distribution data in the field. Due to the strong explanation of counterfactual, we leverage counterfactual to disambiguate the semantic overlap between labels.

Chapter 7

Conclusion and Discussion

7.1 Conclusion

With the explosive growth of text data and computer power, data-driven algorithms, e.g., deep learning, enable breakthrough advances in various NLP tasks. Most of this text data is unstructured and unlabeled. However, RE tasks rely on a large amount of annotated data for training a robust RE model. Since it is time-consuming and labor-intensive to manually annotate the structured triple set from plain text, the DS method introduces existing KGs to automatically annotate training data with a strong assumption. The strong assumption hypothesizes that two named entities in various sentences represent a consistent relationship. Such arbitrary assumption is inevitably faced with two major challenges: 1) the wrong labeling problem, and 2) the long-tail relations. Though the multi-instance learning framework and the selective attention network are proposed to tackle the wrong labeling problem in DSRE, the ontology knowledge between sentences, named entities, and hierarchical relations is not been fully leveraged.

In this research, we first focus on tackling the two major challenges in DSRE with sufficient labeled training data. Then, we also try to solve the RE task in the data-scarce setting, a.k.a., the few-shot setting.

We first propose a selective gate framework with the entity-aware embedding approach for DSRE. Compared to existing work, our novel entity-aware embedding approach is able to dynamically integrate entity information into each word embedding for generating more expressively-powerful representations. Based on the specifically designed embedding layer, we develop a lightweight self-attention mechanism to make up

for the lack of PCNN. Thus, the sentence distributed representation contains both local structured information and the long dependencies between words. Based on the preceding features, we replace the former selective attention network with a selective gate to aggregate sentence-level into bag-level.

Secondly, we mainly focus on addressing long-tail relations. Unlike former works that set up isolated multi-granular relation embedding as the query of attention network, we propose collaborating relation-augmented attention network in the hierarchy. With the relation-augmented process, the sentence representation is also fulfilled with the relational information. In the relational hierarchy, such information will be transformed from data-rich relations into long-tail ones, which is far more efficient than former works.

Thirdly, we introduce extra ontology knowledge, type information of named entities, from existing KGs. Though former works leverage relational hierarchy to share the information with the long-tail relations, the coarse-grained relation is hard to align with the sentence's semantics. Thus, the type of information is a direct and promising way to overcome this challenge while former works ignore this potential of explicit structured type information.

Finally, we also research in RE task with less truly labeled training data, a.k.a., few-shot setting. There will be semantic overlap between the relations of the RE task. Many works leverage the strong ability of PLMs and fine-tuning paradigms to stimulate the knowledge in PLMs. To accurately predict the relation with less training data However, the PLMs are hard to fine-tune with insufficient labeled training data. Thus, inspired by the research in social science humans rely on exploring contrastive attributes between the object factual and other potential counterfactuals, rather than the explicit evidence of the objective factual to classify. Hence, we propose a counterfactual contrastive prompt-tuning approach to extract the contrastive attributes for tackling the RE task and other many-class classification tasks under the few-shot setting.

7.2 Discussion and Future Work

In this section, we will discuss the implications of our findings, the limitations of the current study, and the potential future research directions in the field of the RE task. The rapid advancement of generative pre-trained models, like GPT-3, has notably enhanced the efficacy of RE tasks. Meanwhile, with the rapid increase in computational power and the expansion of model sizes, an increasing amount of knowledge is being encapsulated within large language models, enabling them to adeptly handle a variety of tasks. However, for Relation Extraction (RE), there remain some challenges. For instance, challenges include how to counter adversarial samples, how to enhance the utility of large language models through integration with knowledge graphs, and how to broaden the application scope of RE tasks by incorporating multi-modal information among others. We will explore several possible research directions and opportunities for future work.

1. The effectiveness of RE tasks significantly fluctuates across various domains. Enhancing their resilience, particularly against adversarial samples and attacks, necessitates a focus on domain adaptation and transfer learning. By harnessing cross-domain capabilities, we can optimize the use of pre-trained models for domain-specific RE tasks while bolstering the models' robustness to diverse attack types. This cross-domain prowess is crucial for understanding and adapting to the unique characteristics of data in different fields, ensuring stable and effective performance even against sophisticated or malicious adversarial samples.

2. By integrating large language models with KGs, we can enhance the authenticity of the output, mitigating certain levels of illusion. Simultaneously, the objective truths inherent in knowledge graphs can serve as background knowledge, assisting large language models in making accurate judgments during relation extraction tasks. This integration facilitates a degree of knowledge transfer, enabling the models to apply learned concepts and relationships to new and diverse contexts.

3. Most existing relation extraction research focuses on text data. However, relation extraction in real-world applications may require information from various types of data, such as images, videos, and audio. Integrating multi-modal data into the models could potentially broaden the application scenarios of relation extraction.

4. Future work should also emphasize ontological knowledge learning in proprietary domains, where data privacy is paramount. To establish effective models capable of capturing limited, domain-specific data, leveraging federated learning becomes essential. Federated learning allows for the construction of robust models by training across multiple decentralized devices or servers holding private data, without exchanging them. Thus, it ensures data privacy while aggregating and improving ontological knowledge capture. Some existing studies [113, 114, 115, 116, 117, 118, 119, 120] have successfully begun to construct such neutral models, demonstrating the feasibility and potential of this approach. Advancing these techniques will enable more precise and secure extraction of ontological knowledge in specialized fields, addressing the unique challenges of privacy-sensitive domains.

Bibliography

[1] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1753–1762. The Association for Computational Linguistics, 2015.

[2] Wenyuan Zeng, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Incorporating relation paths in neural relation extraction. *arXiv preprint arXiv:1609.07479*, 2016.

[3] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28, 2014.

[4] Xu Han, Pengfei Yu, Zhiyuan Liu, Maosong Sun, and Peng Li. Hierarchical relation extraction with coarse-to-fine grained attention. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2236–2245. Association for Computational Linguistics, 2018.

[5] Ningyu Zhang, Shumin Deng, Zhanlin Sun, Guanying Wang, Xi Chen, Wei Zhang, and Huajun Chen. Long-tail relation extraction via knowledge graph embeddings and graph convolution networks. In Jill Burstein, Christy Doran, and

Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3016–3025. Association for Computational Linguistics, 2019.

[6] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. Disan: Directional self-attention network for rnn/cnn-free language understanding. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5446–5455. AAAI Press, 2018.

[7] Ashish Vaswani, Shazeer, Noam, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *The Neural Information Processing Systems*, 2017.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[9] Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. PTR: prompt tuning with rules for text classification. *CoRR*, abs/2105.11259, 2021.

[10] Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. Few-nerd: A few-shot named entity recognition dataset. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3198–3213. Association for Computational Linguistics, 2021.

[11] Zhi-Xiu Ye and Zhen-Hua Ling. Distant supervision relation extraction with intra-bag and inter-bag attentions. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2810–2819. Association for Computational Linguistics, 2019.

[12] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. *SIGIR Forum*, 51(2):202–208, 2017.

[13] Kiran Adnan and Rehan Akbar. An analytical study of information extraction from unstructured and multidimensional big data. *J. Big Data*, 6:91, 2019.

[14] Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. A joint neural model for information extraction with global features. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7999–8009. Association for Computational Linguistics, 2020.

[15] Alan Akbik, Tanja Bergmann, and Roland Vollgraf. Pooled contextualized embeddings for named entity recognition. In Jill Burstein, Christy Doran, and

Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 724–728. Association for Computational Linguistics, 2019.

[16] Ralph Grishman and Beth Sundheim. Message understanding conference- 6: A brief history. In *16th International Conference on Computational Linguistics, Proceedings of the Conference, COLING 1996, Center for Sprogteknologi, Copenhagen, Denmark, August 5-9, 1996*, pages 466–471, 1996.

[17] Archana Goyal, Vishal Gupta, and Manish Kumar. Recent named entity recognition and classification techniques: A systematic review. *Comput. Sci. Rev.*, 29:21–43, 2018.

[18] Rodrigo Agerri and German Rigau. Robust multilingual named entity recognition with shallow semi-supervised features (extended abstract). In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 4965–4969. ijcai.org, 2017.

[19] Alexandre Passos, Vineet Kumar, and Andrew McCallum. Lexicon infused phrase embeddings for named entity resolution. In Roser Morante and Wen-tau Yih, editors, *Proceedings of the Eighteenth Conference on Computational Natural Language Learning, CoNLL 2014, Baltimore, Maryland, USA, June 26-27, 2014*, pages 78–86. ACL, 2014.

[20] Yang Li, Tao Shen, Guodong Long, Jing Jiang, Tianyi Zhou, and Chengqi Zhang. Improving long-tail relation extraction with collaborating relation-augmented attention. In Donia Scott, Núria Bel, and Chengqing Zong, editors, *Proceedings of*

the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020, pages 1653–1664. International Committee on Computational Linguistics, 2020.

[21] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In Jan Hajic and Junichi Tsujii, editors, *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 2335–2344. ACL, 2014.

[22] Sauro Menchetti, Fabrizio Costa, Paolo Frasconi, and Massimiliano Pontil. Wide coverage natural language processing using kernel methods and neural networks for structured data. *Pattern Recognit. Lett.*, 26(12):1896–1906, 2005.

[23] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing [review article]. *IEEE Comput. Intell. Mag.*, 13(3):55–75, 2018.

[24] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[25] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119, 2013.

[26] Razvan C. Bunescu and Raymond J. Mooney. Subsequence kernels for relation extraction. In *Advances in Neural Information Processing Systems 18 [Neural*

Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada], pages 171–178, 2005.

[27] Thien Huu Nguyen and Ralph Grishman. Relation extraction: Perspective from convolutional neural networks. In Phil Blunsom, Shay B. Cohen, Paramveer S. Dhillon, and Percy Liang, editors, *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, VS@NAACL-HLT 2015, June 5, 2015, Denver, Colorado, USA*, pages 39–48. The Association for Computational Linguistics, 2015.

[28] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *Journal of machine learning research*, 3(Feb):1083–1106, 2003.

[29] Eugene Agichtein and Luis Gravano. *Snowball*: extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM Conference on Digital Libraries, June 2-7, 2000, San Antonio, TX, USA*, pages 85–94. ACM, 2000.

[30] Sergey Brin. Extracting patterns and relations from the world wide web. In Paolo Atzeni, Alberto O. Mendelzon, and Giansalvatore Mecca, editors, *The World Wide Web and Databases, International Workshop WebDB'98, Valencia, Spain, March 27-28, 1998, Selected Papers*, volume 1590 of *Lecture Notes in Computer Science*, pages 172–183. Springer, 1998.

[31] Oren Etzioni, Michael J. Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artif. Intell.*, 165(1):91–134, 2005.

[32] Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. Open information extraction: The second generation. In Toby Walsh,

editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 3–10. IJCAI/AAAI, 2011.

[33] Fei Wu and Daniel S. Weld. Open information extraction using wikipedia. In Jan Hajic, Sandra Carberry, and Stephen Clark, editors, *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 118–127. The Association for Computer Linguistics, 2010.

[34] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In Keh-Yih Su, Jian Su, and Janyce Wiebe, editors, *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, pages 1003–1011. The Association for Computer Linguistics, 2009.

[35] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In José L. Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag, editors, *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III*, volume 6323 of *Lecture Notes in Computer Science*, pages 148–163. Springer, 2010.

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

[37] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016.

[38] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In Jason Tsong-Li Wang, editor, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250. ACM, 2008.

[39] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *OpenAI Research*, 2018.

[40] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[41] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[42] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

[43] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In Anna Korhonen, David R. Traum, and Lluís

Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3651–3657. Association for Computational Linguistics, 2019.

[44] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics, 2018.

[45] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 328–339. Association for Computational Linguistics, 2018.

[46] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing*

Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

[47] Timo Schick, Helmut Schmid, and Hinrich Schütze. Automatically identifying words that can serve as labels for few-shot text classification. In Donia Scott, Núria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 5569–5578. International Committee on Computational Linguistics, 2020.

[48] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. GPT understands, too. *CoRR*, abs/2103.10385, 2021.

[49] Timo Schick and Hinrich Schütze. It's not just size that matters: Small language models are also few-shot learners. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2339–2352. Association for Computational Linguistics, 2021.

[50] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4222–4235. Association for Computational Linguistics, 2020.

[51] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto

Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3816–3830. Association for Computational Linguistics, 2021.

[52] Zexuan Zhong, Dan Friedman, and Danqi Chen. Factual probing is [MASK]: learning vs. learning to recall. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5017–5033. Association for Computational Linguistics, 2021.

[53] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3045–3059. Association for Computational Linguistics, 2021.

[54] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597. Association for Computational Linguistics, 2021.

[55] Yoon Kim. Convolutional neural networks for sentence classification. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of*

the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 1746–1751. ACL, 2014.

[56] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795, 2013.

[57] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[58] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3837–3845, 2016.

[59] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[60] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015.

[61] Rafal Józefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *CoRR*, abs/1602.02410, 2016.

[62] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[63] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[64] Jinhua Du, Jingguang Han, Andy Way, and Dadong Wan. Multi-level structured self-attentions for distantly supervised relation extraction. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2216–2225. Association for Computational Linguistics, 2018.

[65] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. Qanet: Combining local convolution with global self-attention for reading comprehension. In *The International Conference on Learning Representations*, 2018.

[66] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[67] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3060–3066. AAAI Press, 2017.

[68] Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. *arXiv preprint arXiv:1901.10430*, 2019.

[69] Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. Learning natural language inference using bidirectional LSTM model and inner-attention. *CoRR*, abs/1605.09090, 2016.

[70] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. Tensorized self-attention: Efficiently modeling pairwise and global dependencies together. In *NAACL*, pages 1256–1266, 2019.

[71] Tianyu Liu, Kexiang Wang, Baobao Chang, and Zhifang Sui. A soft-label method for noise-tolerant distantly supervised relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1790–1795, 2017.

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

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

[74] Yang Li, Guodong Long, Tao Shen, Tianyi Zhou, Lina Yao, Huan Huo, and Jing Jiang. Self-attention enhanced selective gate with entity-aware embedding for distantly supervised relation extraction. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of*

Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 8269–8276. AAAI Press, 2020.

[75] Samy Bengio, Jason Weston, and David Grangier. Label embedding trees for large multi-class tasks. In John D. Lafferty, Christopher K. I. Williams, John Shawe-Taylor, Richard S. Zemel, and Aron Culotta, editors, *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 163–171. Curran Associates, Inc., 2010.

[76] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. Bi-directional block self-attention for fast and memory-efficient sequence modeling. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[77] Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.

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

[79] Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha P. Talukdar. RESIDE: improving distantly-supervised neural relation extraction using side information. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on*

Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, pages 1257–1266. Association for Computational Linguistics, 2018.

[80] Linmei Hu, Luhao Zhang, Chuan Shi, Liqiang Nie, Weili Guan, and Cheng Yang. Improving distantly-supervised relation extraction with joint label embedding. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3819–3827. Association for Computational Linguistics, 2019.

[81] Zhendong Chu, Haiyun Jiang, Yanghua Xiao, and Wei Wang. Insrl: A multi-view learning framework fusing multiple information sources for distantly-supervised relation extraction. *CoRR*, abs/2012.09370, 2020.

[82] Shuang Chen, Jinpeng Wang, Feng Jiang, and Chin-Yew Lin. Improving entity linking by modeling latent entity type information. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7529–7537. AAAI Press, 2020.

[83] Denis Krompaß, Stephan Baier, and Volker Tresp. Type-constrained representation learning in knowledge graphs. In Marcelo Arenas, Óscar Corcho, Elena Simperl, Markus Strohmaier, Mathieu d'Aquin, Kavitha Srinivas, Paul Groth, Michel Dumontier, Jeff Heflin, Krishnaprasad Thirunarayan, and Steffen Staab, editors, *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I*, volume 9366 of *Lecture Notes in Computer Science*, pages 640–655. Springer,

2015.

[84] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 809–816. Omnipress, 2011.

[85] Xin Li and Dan Roth. Learning question classifiers. In *ACL*, 2002.

[86] Bo Chen, Xiaotao Gu, Yufeng Hu, Siliang Tang, Guoping Hu, Yueting Zhuang, and Xiang Ren. Improving distantly-supervised entity typing with compact latent space clustering. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2862–2872. Association for Computational Linguistics, 2019.

[87] Trieu H. Trinh and Quoc V. Le. A simple method for commonsense reasoning. *CoRR*, abs/1806.02847, 2018.

[88] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. Language models as knowledge bases? In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2463–2473. Association for Computational Linguistics, 2019.

[89] Joe Davison, Joshua Feldman, and Alexander M. Rush. Commonsense

knowledge mining from pretrained models. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1173–1178. Association for Computational Linguistics, 2019.

[90] Albert Webson and Ellie Pavlick. Do prompt-based models really understand the meaning of their prompts? In Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2300–2344. Association for Computational Linguistics, 2022.

[91] Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. Knowledgeable or educated guess? revisiting language models as knowledge bases. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 1860–1874. Association for Computational Linguistics, 2021.

[92] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.*, 267:1–38, 2019.

[93] Alon Jacovi, Swabha Swayamdipta, Shauli Ravfogel, Yanai Elazar, Yejin Choi, and Yoav Goldberg. Contrastive explanations for model interpretability. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural*

Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pages 1597–1611. Association for Computational Linguistics, 2021.

[94] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 15750–15758. Computer Vision Foundation / IEEE, 2021.

[95] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 35–45. Association for Computational Linguistics, 2017.

[96] Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1558–1569. Association for Computational Linguistics, 2020.

[97] George Stoica, Emmanouil Antonios Platanios, and Barnabás Póczos. Re-tacred: Addressing shortcomings of the TACRED dataset. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13843–13850. AAAI Press, 2021.

[98] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In Eneko Agirre, Lluís Màrquez, and Richard Wicentowski, editors, *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions, SEW@NAACL-HLT 2009, Boulder, CO, USA, June 4, 2009*, pages 94–99. Association for Computational Linguistics, 2009.

[99] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015.

[100] Yuhao Zhang, Peng Qi, and Christopher D. Manning. Graph convolution over pruned dependency trees improves relation extraction. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2205–2215. Association for Computational Linguistics, 2018.

[101] Matthew E. Peters, Mark Neumann, Robert L. Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. Knowledge enhanced contextual word representations. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 43–54. Association for Computational Linguistics, 2019.

[102] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and

Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguistics*, 8:64–77, 2020.

[103] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. LUKE: deep contextualized entity representations with entity-aware self-attention. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6442–6454. Association for Computational Linguistics, 2020.

[104] Ganqu Cui, Shengding Hu, Ning Ding, Longtao Huang, and Zhiyuan Liu. Prototypical verbalizer for prompt-based few-shot tuning. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 7014–7024. Association for Computational Linguistics, 2022.

[105] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035, 2019.

[106] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. Association for Computational Linguistics, 2020.

[107] Alexis Ross, Ana Marasovic, and Matthew E. Peters. Explaining NLP models via minimal contrastive editing (mice). In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 3840–3852. Association for Computational Linguistics, 2021.

[108] Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating models' local decision boundaries via contrast sets. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1307–1323. Association for Computational Linguistics, 2020.

[109] Bhargavi Paranjape, Julian Michael, Marjan Ghazvininejad, Hannaneh Hajishirzi, and Luke Zettlemoyer. Prompting contrastive explanations for commonsense reasoning tasks. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli,

editors, *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 4179–4192. Association for Computational Linguistics, 2021.

[110] Zhu Zhang, Zhou Zhao, Zhijie Lin, Jieming Zhu, and Xiuqiang He. Counterfactual contrastive learning for weakly-supervised vision-language grounding. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[111] Divyansh Kaushik, Eduard H. Hovy, and Zachary Chase Lipton. Learning the difference that makes A difference with counterfactually-augmented data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[112] Linyi Yang, Jiazheng Li, Padraig Cunningham, Yue Zhang, Barry Smyth, and Ruihai Dong. Exploring the efficacy of automatically generated counterfactuals for sentiment analysis. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 306–316. Association for Computational Linguistics, 2021.

[113] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8432–8440, 2022.

[114] Guodong Long, Ming Xie, Tao Shen, Tianyi Zhou, Xianzhi Wang, and Jing Jiang. Multi-center federated learning: clients clustering for better personalization. *World Wide Web*, 26(1):481–500, 2023.

[115] Yue Tan, Guodong Long, Jie Ma, Lu Liu, Tianyi Zhou, and Jing Jiang. Federated learning from pre-trained models: A contrastive learning approach. *Advances in Neural Information Processing Systems*, 35:19332–19344, 2022.

[116] Fengwen Chen, Guodong Long, Zonghan Wu, Tianyi Zhou, and Jing Jiang. Personalized federated learning with graph. *arXiv preprint arXiv:2203.00829*, 2022.

[117] Jie Ma, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. Structured federated learning through clustered additive modeling. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[118] Peng Yan and Guodong Long. Personalization disentanglement for federated learning. *arXiv preprint arXiv:2306.03570*, 2023.

[119] Guodong Long, Tao Shen, Yue Tan, Leah Gerrard, Allison Clarke, and Jing Jiang. Federated learning for privacy-preserving open innovation future on digital health. In *Humanity Driven AI: Productivity, Well-being, Sustainability and Partnership*, pages 113–133. Springer, 2021.

[120] Guodong Long, Yue Tan, Jing Jiang, and Chengqi Zhang. Federated learning for open banking. In *Federated Learning: Privacy and Incentive*, pages 240–254. Springer, 2020.