Refining latent multi-view graph for relation extraction

Xue, Fuzhao

2021

Xue, F. (2021). Refining latent multi-view graph for relation extraction. Master's thesis, Nanyang Technological University, Singapore. <https://hdl.handle.net/10356/151541>

<https://hdl.handle.net/10356/151541>

<https://doi.org/10.32657/10356/151541>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 16:25:33 SGT

Refining Latent Multi-View Graph for Relation Extraction

Xue Fuzhao

Main Supervisor: Prof. Chng Eng Siong

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Master of Engineering

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

23/03/2021

.....

Date

Xue Fuzhao

.....

Xue Fuzhao

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

23/03/2021

.....

Date

.....

Prof. Chng Eng Siong

Authorship Attribution Statement

This thesis contains material from 1 paper published in the following paper accepted at a conference in which I am listed as an author.

Chapter 3 and Chapter 4 are published as [Fuzhao Xue, Aixin Sun, Hao Zhang, Eng-Siong Chng](#) "GDPNet: Refining Refining Latent Multi-View Graph for Relation Extraction" Association for the Advancement of Artificial Intelligence (AAAI) 2021

The contributions of the co-authors are as follows:

- I came up with the idea and did all implementations. I also wrote the manuscript draft.
- A/Prof Sun commented on my idea. He helped me to make the experimental plan. The manuscript was revised by A/Prof Sun and Hao Zhang.
- A/Prof Chng provided the initial project direction and edited the manuscript drafts.

23/03/2021

.....

Date

XueFuzhao

.....

Xue Fuzhao

Acknowledgements

I would like to express my sincere appreciation to my main supervisor Dr. Chng Eng Siong and my co-supervisor Dr. Sun Aixin. Under their kind supervision, I have learned not only substantial knowledge about information extraction, text mining and audio classification, but also an inspiring and down-to-earth research attitude. I would also like to express my greatest gratitude to my parents for providing me with unfailing support and continuous encouragements throughout my entire life.

I would also like to acknowledge my friends and colleagues: Mr. Zhang Hao, Mr. Andrew Koh Jin Jie, Mr. Lim Zhi Hao, Ms. Hou Nana, Dr. Xu Haihua, Mr. Ni Jinjie, Mr. Zhou Haicang, for their great support and encouragement throughout my graduate study period in SCSE NTU.

Abstract

Relation extraction (RE) is to predict the relation type of two entities that are mentioned in a piece of text, *e.g.*, a sentence or a dialogue. According to various text input formats, Relation extraction can be divided into different categories, such as sentence-level RE, document-level RE and dialogue-level RE. In this thesis, we focus on addressing sentence-level RE and dialogue-level RE with a unified model. For both sentence-level RE and dialogue-level RE, the main challenge arise when the given text is long. It is difficult to identify indicative words for the relation prediction.

Recent advances on RE tasks are from BERT-based models. These models predict relation by sentence-level semantic representation. Hierarchical representations are missing in BERT-based models so that the words weakly related to the target relation induce extra noise. Therefore, in this thesis, we propose a unified model, GDPNet (Gaussian Dynamic Time Warping Pooling Net), to improve relation extraction by identifying indicative words. GDPNet contains a BERT module and a Graph Module. In the BERT module, we obtain token representations from the raw text. Then, in the Graph Module, we propose to construct a latent multi-view graph to capture various possible relationships among tokens. We then refine this graph to select important words for relation prediction. Finally, the representation of the refined graph and the BERT-based sequence representation are concatenated for relation extraction. Specifically, in our proposed GDPNet, we utilize Gaussian Graph Generator (GGG) to generate edges of the multi-view graph. The graph is then refined by Dynamic Time Warping Pooling (DTWPool).

On DialogRE and TACRED, we show that GDPNet achieves the best performance on dialogue-level RE, and comparable performance with the state-of-the-arts on sentence-level RE. We also perform a quantitative study to show that our model is capable of finding indicative words in an unsupervised manner.

Contents

Acknowledgements	ix
Abstract	xi
List of Figures	xv
List of Tables	xvii
1 Introduction	1
1.1 Motivation	3
1.2 Method	5
1.3 Contributions	7
1.4 Organization of the Report	7
2 Literature Review	9
2.1 Background	9
2.1.1 Problem Definition	9
2.1.2 Categories and Focuses	10
2.2 Sentence-level relation extraction	11
2.2.1 Datasets	11
2.2.2 Approaches	14
2.2.2.1 RNN-Based Models	14
2.2.2.2 Graph-Based Models	15
2.2.2.3 BERT-Based Models	16
2.2.2.4 State-of-the-Art	17
2.3 Dialogue-level relation extraction	18
2.3.1 Datasets	18
2.3.2 Approaches	19
2.3.2.1 RNN-Based Models	19
2.3.2.2 Graph-Based Models	20
2.3.2.3 BERT-Based Models	21
2.3.2.4 State-of-the-Art	22
2.4 Challenges	23

2.5	Summary	24
3	Approach	25
3.1	Potential Problem	25
3.2	Proposed Solution	26
3.2.1	Preliminary	26
3.2.1.1	Problem Formulation	26
3.2.1.2	Multi-View Graph	27
3.2.1.3	SoftDTW	28
3.2.2	GDPNet	30
3.2.2.1	BERT Module	31
3.2.2.2	Graph Module	34
3.2.2.3	Classifier	42
3.3	Summary	42
4	Experiments	43
4.1	Dialogue-level Relation Extraction	44
4.1.1	Datasets	44
4.1.2	Baseline Models	44
4.1.3	Experimental Setup	44
4.1.4	Results on DialogRE	45
4.1.5	Case Study	47
4.1.6	Ablation Study	48
4.2	Sentence-level Relation Extraction	49
4.2.1	Datasets	49
4.2.2	Baseline Models	50
4.2.3	Experimental Setup	50
4.2.4	Results on TACRED	51
4.2.5	Results on TACRED-Revisit	52
4.3	Quantitative Analysis	53
4.4	Summary	54
5	Conclusion and Future Work	55
5.1	Future Work	56
	List of Author's Awards, Patents, and Publications	59
	Bibliography	61

List of Figures

3.1	Comparison between homogeneous graph and multi-view graph . .	27
3.2	The overall architecture of the proposed GDPNet. Entities E_1 and E_2 are single-token entities in the illustration.	30
3.3	Input format of BERT and BERTs	32
3.4	The overall architecture of the graph module.	34
4.1	GDPNet and BERTs on different dialogue lengths.	46

List of Tables

1.1	An example from DialogRE dataset [1].	3
2.1	Different categories of relation extraction tasks and their corresponding input text format.	10
2.2	TACRED statistics per split. About 79.5% of the examples are labeled as no relation (Negative Examples). [2]	13
2.3	Performance of the existing models on TACRED.	17
2.4	Performance of all models on DialogRE. σ denotes the standard deviation computed from five runs of each model.	22
4.1	Hyperparameter settings on DialogRE.	45
4.2	Performance of all models on DialogRE. σ denotes the standard deviation computed from five runs of each model.	46
4.3	Text of the first dialogue in test set of DialogRE. The tokens in bold are selected by GDPNet.	47
4.4	An ablation study on GDPNet model. For models without GGG, we replaced GGG by multi-head attention to initialize the edges of the multi-view graph. For the model without DTWloss, we remove the second term in Equation 3.25. All DTWPool layers are removed for the models without DTWPool.	48
4.5	Hyperparameter settings on TACRED.	51
4.6	Performance of all models on TACRED.	51
4.7	Performance of all models on TACRED-Revisit. we run the released code if available, and mark results obtained by asterisk(*). For the models without reported performance on TACRED-Revisit, we run the released code if available.	52
4.8	Percentage (%) of the tokens selected in the final graph from sequence, <i>e.g.</i> an entire dialogue in DialogRE or the whole sentence in TACRED. Non-repetitive tokens are the tokens that appear only once in the sequence; repetitive tokens appear two or more times in the sequence. Trigger tokens are key tokens annotated in DialogRE.	53

Chapter 1

Introduction

In the past ten years, with the development of information technology, information on the Internet has grown at an alarming rate. To make use of the information, natural language plays an important role. To acquire knowledge from the unstructured text effectively, researchers designed many Information Extraction tasks, such as named entity recognition [3], slot filling [4] and relation extraction [2]. Among these tasks, Relation extraction (RE) is one of the most fundamental and widely used Information Extraction tasks. By extracting relations among entities from the amount of text, we can obtain the structured data, which is beneficial to the downstream tasks like Knowledge Graph [5], Recommendation System [6] and Q&A [7]. For instance, consider a knowledge-based Q&A system, each relationship extracted from dialogue can be used to connect two entities in the existing knowledge base, which means the relation represents one edge between two nodes in the knowledge graph. We can then employ the updated knowledge graph with more complex relations to improve the Q&A system. Relation extraction can also be used in smart home. Detecting the relation between two speakers in the family during conversation can provide more accurate personalized service for users. In this thesis, we focus on relation extraction and we expect to adapt our model to improve other downstream tasks in the future.

What is the research focus of this thesis? Relation extraction can be mainly divided into sentence-level RE [2], document-level RE [8], dialogue-level RE [1], and few-shot RE [9]. We focus on sentence-level RE and dialogue-level RE in this thesis. The sentence-level RE is the most simple and fundamental relation extraction

task. For the sentence-level RE, one single sentence and two mentioned entities are given, and we are to predict the relation type between these two mentioned entities. The input text of document-level RE is a piece of the document instead of one single sentence. It is noticeable that the entity pairs are not given to classify the relation type in document-level RE. Instead, in document-level RE, we only know the entities, and we are required to predict the relations of all combinations of the entities. The input text of dialogue-level RE is a piece of dialogue, and more pairs of entities are related to speakers. Please note that the argument pairs (entity pairs) are given in dialogue-level RE, which is similar to sentence-level RE but different from document-level RE. The few-shot RE has a different focus, which works on training models by few-shot learning instead of using a large-scale annotated dataset. In this thesis, we focus on sentence-level RE and dialogue-level RE and leave the document-level RE as our future work. Few-shot RE, as we mentioned above, has a different focus with this thesis, so we will not further discuss the few-shot RE in this work.

Problem Definition. Given two entities and a piece of text where the two entities are mentioned, the task of relation extraction is to predict the semantic relation between the two entities. The piece of text serves as the context for the prediction, which can be a short sentence, a long sentence, or even a dialogue. We use an example from TACRED [2] dataset to illustrate the RE task. In this example, we are interested in predicting the relation type between two entities: “*Cathleen P. Black*” and “*chairwoman*”. Based on a sentence: “*Carey will succeed Cathleen P. Black, who held the position for 15 years and will take on a new role as chairwoman of Hearst Magazines, the company said*”, we aim to predict the relation between these two entities to be “per:title”. Note that, the relation types in relation extraction tasks are predefined. For instance, TARCED dataset defines 41 types and a special “no relation” type if a predicted relation is not covered in the 41 predefined types.

How useful are the research topics of this thesis? One research focus of this thesis, sentence-level RE, is a classical natural language understanding task, which has been widely used to evaluate various language models, such as BERT [10], SpanBERT [11], KnowBERT [12], KEPLER [13] and K-ADAPTER [14]. In addition, sentence-level RE can be used to improve other downstream tasks like Knowledge

TABLE 1.1: An example from DialogRE dataset [1].

S1:	Hey Pheeb.		
S2:	Hey!		
S1:	Any sign of your brother ?		
S2:	No, but he is always late.		
S1:	I thought you only met him once?		
S2:	Yeah, I did. I think it sounds y’know big sistery, y’know, ‘Frank’s always late.’		
S1:	Well relax, he’ll be here.		
	Argument Pair	Trigger word	Relation type
R1	(Frank, S2)	brother	per:siblings
R2	(S2, Pheeb)	none	per:alternate names

base Q&A [15]. For sentence-level RE, there are also various real-world applications. For instance, sentence-level biomedical relation extraction [16] is an important medical text mining task that aims to automatically extract valuable biomedical interactions between protein-protein, drug-drug, or chemical disease. However, in more cases, the text format is more complex than the setting of sentence-level RE. Two given entities may not appear in the same sentence, and the context may also change due to different data sources. For example, when the text is from a newspaper or book, the input text of relation extraction would be a document. The context would be a dialogue if we obtain data from daily conversation. Dialogue-level RE is designed for extracting relation from the dialogue text, and it has many potential application scenarios. Relation between two entities within a dialogue can be used to generate a dynamic knowledge graph. Then, we can adopt the dynamic knowledge graph to improve ChatBot [17] and dialogue system[18], and even provide personal service to family members by constructing a family member graph.

1.1 Motivation

Data and Problem Analyze. The motivation of this thesis comes from the observation from data. This thesis uses a data sample (see Table 1.1) from a widely used dialogue-level RE dataset, DialogRE, to show our motivation. Observe from the example in Table 1.1, only a few words in the given context are related to the semantic relation between the two entities. This is a shorter dialogue actually.

In a dialogue with more sentences or a single yet much longer sentence, most of the remaining words in the given text are less relevant to the prediction. For instance, we can observe that Relation 1 (R1) can be easily predicted based on a trigger word (*e.g.*, “brother”), despite the long conversation between S1 and S2. DialogRE dataset even provides trigger word annotation, which is the smallest span of text that most clearly indicates the existence of the relation between two arguments. For both sentence-level RE and dialogue-level RE, the informative words are sparse in a longer text sequence. The remaining words include more semantic information that is unrelated to the target relation. In other words, these remaining words induce extra noise for prediction. Therefore, in this thesis, we are aiming at improving relation extraction models by identifying these sparse informative words when the context is long.

Existing work. In relation extraction, the existing works adopted handcrafted features in the early stage. Various external tools and manually designing kernels [19] are widely used to improve performance. However, the classification models relying on the high-level lexical and syntactic features obtained from external tools and manually designing kernels suffer from error propagation of the tools. Recently, deep neural networks have outperformed previous models using handcraft features. The existing neural models in RE mainly include RNN-based models [20, 21], graph-based models and BERT-based models. Among RNN-based models, Att-BiLSTM [21] use attention mechanism and Bidirectional Long Short-Term Memory Networks to capture the long dependency. PA-LSTM [2] employs a position-aware attention mechanism over LSTM outputs. SA-LSTM [22] aims to incorporate segment information into neural relation extractor by attention. There are also a few RNN-based models like SDP-LSTM [20] and Tree-LSTM [23] incorporating dependency trees into RNN-based models to consider syntactic information. These approaches considering dependency tree can also be classified into graph-based models because they use the dependency tree as a graph to model the relation. C-GCN[24] and C-AGGCN[25] use a similar framework as above, which means they use RNN-based models to model the context, and Graph-based models are used to capture the long dependency and syntactic information. However, both Graph-based models and RNN-based models are weak in modeling semantics comparing with BERT-based models [10]. Like other natural language understanding tasks, BERT-based models achieved state-of-the-art performance on relation

extraction. Self-supervised training from huge unstructured text provides outstanding performance to reason and model semantic to BERT. In sentence-level RE, SpanBERT [11] achieved the best performance when no extra training data are applied. SpanBERT uses a modified training strategy to mask a span of tokens during pretraining to capture more long dependency semantic. BERTs [1] is the best model in dialogue-level RE, which takes speaker-related information into consideration. However, in both SpanBERT and BERTs, hierarchical representations are missing during prediction. As we mentioned above, the informative tokens are sparse and most tokens are weakly related to the target relation. Therefore, these unrelated tokens induce extra noise for BERT-based models, so they limit the performance of the BERT-based models.

1.2 Method

Introduce our method. This thesis aims to improve relation extraction by refining latent multi-view graph. We focus on sentence-level RE and dialogue-level RE. Different from the most existing works, we propose one single model, GDPNet, which can handle these two tasks with a unified structure. Generally, we combine the BERT-based model and graph-based model to obtain a set of hierarchical representations. Specially, we first use the BERT-based model named as BERT module in this thesis to obtain token representations. The input of the BERT-based model is a token sequence, which starts with "[CLS]" tokens and ends with "[SEP]" tokens. The middle part of the input is a piece of text. In sentence-level RE, the middle part is a single sentence, and it is a piece of dialogue in dialogue-level RE. Then, in the BERT module, the input, a token sequence, is fed into one BERT-based model to obtain token representations¹. Then one graph module is employed on the top of the BERT module. The main contributions of this thesis are covered by the graph module. To construct a graph to model the asymmetrical relation, we first propose Gaussian Graph Generator (GGG). GGG adopts Gaussian distribution and Kullback-Leibler (KL) divergence to generate the edges of latent multi-view graph by computing semantic dissimilarity. The latent graph generated is a directed graph due to the asymmetrical nature of KL divergence. Each token representation corresponds to one node in the multi-view

¹In our thesis, for a fair comparison, we use SpanBERT as backbone, and BERTs is applied in dialogue-level RE.

graph, and each node generates multiple Gaussian distributions. In other words, the number of nodes in our graph module is equal to the number of tokens in the input of BERT module. Given the large fully-connected latent multi-view graph generated from token representations by GGG, our next goal is to refine this large graph due to the sparsity of informative tokens. In order to obtain a smaller task-specific graph, DTWPool is proposed to refine the latent multi-view graph. Different from most graph learning tasks, our graph is a latent graph learned from text, which means the nodes in this graph include important sequential information. Instead of the graph pooling methods shuffling all nodes, The proposed DTWPool uses SoftDTW loss [26] to preserve more sequential information when removing amounts of nodes. We operate the graph convolution and DTWPool iteratively in the graph module, so we can obtain a small task-specific graph with informative tokens (nodes). The first [CLS] token representation, which is widely used as the final representation in sequence classification task, and the task-specific graph are concatenated together and fed into a linear layer to predict the relation type in our GDPNet.

Analyze and evaluate our method. Generally, unlike the existing models in sentence-level RE and dialogue-level RE, our GDPNet combines the BERT-based model and graph-based model. The BERT-based model is used to obtain semantic representations and graph-based model can model the asymmetrical relation in a hierarchical manner. Specially, The proposed GGG in our GDPNet provides a novel way to generate edges for the latent directed multi-view graph. Compared with existing graph pooling methods, the DTWPool proposed is designed for the latent graph learned from sequence, which can preserve more sequential information in the nodes during graph pooling. In this thesis, we evaluate our GDPNet on relation extraction tasks, dialogue-level RE, and sentence-level RE. First, in the dialogue-level RE, we conduct experiments on DialogRE [1] dataset, and our model achieved the best performance. Then, we further evaluate GDPNet on TACRED [2] and TACRED-Revisit [27], two sentence-level RE datasets. We can see that GDPNet achieved comparable performance without any external training resources. In addition, we perform a quantitative study to show that our model is capable of finding indicative words in an unsupervised manner.

1.3 Contributions

Throughout the investigation of this thesis, a novel relation extraction model, GDPNet, is proposed to identify sparse informative tokens to improve relation extraction. We summarize our key contributions as follows:

- We propose a Gaussian Graph Generator (GGG) to initialize edges for latent multi-view graph by measuring KL divergence between different Gaussian distributions of tokens.
- We propose a graph pooling method, DTWPool, to refine the latent multi-view graph learned from text sequence, with a flexible pooling ratio. To the best of our knowledge, this is the first work on multi-view graph pooling.
- We combine GGG and DTWPool to form the GDPNet, and evaluate GDPNet on two benchmark datasets for relation extraction. Experimental results demonstrate the effectiveness of GDPNet against SoTA baselines.

1.4 Organization of the Report

This thesis contains the introduction (this chapter), a literature review, one approach, experiments and the conclusion. The rest of this thesis is organized as follows:

- Chapter Section 2 provides a review of relation extraction approaches in history. Particularly, we review the sentence-level RE and dialogue-level RE tasks, and we discussed the existing problems and potential solutions.
- Chapter Section 3 starts to investigate the specific solution to handle the sparse informative tokens in relation extraction. We first give an overview of our GDPNet. Then a detailed elaboration of Gaussian Graph Generator and DTWPool is provided.
- Chapter Section 4 demonstrates the effectiveness of GDPNet. Our model achieved the best performance on dialogue-level RE and comparable performance on sentence-level RE.

- Chapter Section 5 concludes the thesis and discusses several potential directions for future work.

Chapter 2

Literature Review

In this chapter, we provide a literature review about relation extraction. As a natural language processing (NLP) task, relation extraction plays an important role in Knowledge Acquisition and Information Extraction. Due to the importance of relation extraction, it has been divided into many subtasks, such as sentence-level RE, document-level RE, dialogue-level RE and few-shot RE. Therefore, this chapter begins with Section 2.1. we first introduce the background of relation extraction: the definition, the categories, and our focuses. In this thesis, we mainly focus on sentence-level RE and dialogue-level RE, so we introduce sentence-level RE and dialogue-level RE detailedly in Section 2.2 and Section 2.3, respectively. In both Section 2.2 and Section 2.3, we present the definition, corpus and existing models to review the corresponding task. Last, we discuss the challenges in Section 2.4.

2.1 Background

2.1.1 Problem Definition

Given a piece of text X and two mentioned entities e_s and e_o , relation extraction is to classify the relation type $r \in R$. Usually, the set of relation type R is predefined in the relation extraction dataset. Specially, let $X = \{x_1, x_2, \dots, x_T\}$ be a text sequence with T tokens, where x_t is the token at t^{th} position. To predict relations, we are given two entities, subject entity X_s and object entity X_o . Both X_s and X_o are sub-sequences of X . An entity may contain one or more tokens, *e.g.*,

TABLE 2.1: Different categories of relation extraction tasks and their corresponding input text format.

Class	Text Format	Num of Relations
Sentence-level relation extraction	Sentence	1
Document-level relation extraction	Document	≥ 1
Dialogue-level relation extraction	Dialogue	≥ 1
Few-shot relation extraction	Sentence	1
Medical relation extraction	Sentence	1
Multi-lingual relation extraction	Sentence	1

$X_s = \{x_s, x_{s+1}, \dots, x_{s+m-1}\}$ where s denotes the starting position of X_s in sequence X and m is the number of tokens in X_s . Given X , X_s , and X_o , the goal of relation extraction is to predict the relation $r \in R$ between X_s and X_o .

2.1.2 Categories and Focuses

Relation extraction has been divided into multiple categories due to its importance in NLP. As shown in Table 2.1. These categories mainly include sentence-level RE [2], document-level RE [8], dialogue-level RE [1], few-shot RE [9], medical RE [28, 29] and multi-lingual RE [30]. Sentence-level RE is the most fundamental RE task. The input text sequence X is a sentence in relation extraction. As most relation extraction tasks, two entities e_s and e_o are both given in sentence-level RE. We only predict one relation from one sentence. We can observe that most relation extraction tasks use the single sentence as input text format in Table 2.1. The main differences between document-level RE and sentence-level RE are the input format and the given entity pairs. The input text sequence X is a piece of document instead of one single sentence. In addition, the entity pairs are given in document-level RE, which means all combinations of the given entities are required to be considered. Please note that the number of potential relations n can be larger than one in document-level RE. In other words, we have n pairs of entities in a piece of given document. Dialogue-level RE is very similar to the document-level one. The input text sequence X is a piece of dialogue in dialogue-level relation extraction, and the number of potential relations n can still be larger than one, but the argument (entity) pairs are given in dialogue-level RE. Few-shot RE has a different focus from the relation extraction tasks above. Instead of investigating new text format, few-shot RE argues that the new relation type is always emerging,

so we need to train the relation extraction model with fewer annotated training samples. The input sequence of few-shot RE is the single sentence. Certainly, there are also a few other classes like medical RE [28, 29] and multi-lingual RE [30]. Medical RE focuses on the medical text, so the entity type and relation type have a different domain with other relation extraction tasks.

In this thesis, we mainly focus on sentence-level RE and dialogue-level RE. Therefore, in the following sections of this chapter, Section 2.2 and Section 2.3, we introduce these two tasks in detail.

2.2 Sentence-level relation extraction

Sentence-level relation extraction is to predict one relation between two mentioned entities given one single sentence and the two entities. As we mentioned above, the sentence-level RE is the most basic relation extraction task, which has been treated as a standard NLP task. Therefore, we review sentence-level RE first.

2.2.1 Datasets

In this part, we review five datasets, CoNLL04, ACE2004, ACE2005, SemEval-2010 Task 8 and TACRED, respectively. These five datasets were widely used in the evaluation of sentence-level relation extraction approaches.

CoNLL04 [31] is one of the earliest dataset that can be applied to relation extraction. CoNLL04 dataset is a relatively small scale sentence-level RE dataset. It consists of 1,441 sentences from news articles annotated with four entity types (*i.e.* location, organization, people, and other) and five relation types (*i.e.* works-for, kill, organization-based-in, lives-in, and located-in). This dataset has no overlapping entities. Researchers usually use the three-way split to preprocess the CoNLL04 dataset. The processed dataset contains 910 training, 243 dev, and 288 test sentences. Since CoNLL04 dataset annotated all potential entities and relations, recent works mainly use CoNLL04 dataset to evaluate the joint entity and relation extraction models.

ACE2004 [32] is a multilingual corpus that contains the complete set of English, Arabic and Chinese training data. ACE2004 is also a relatively earlier sentence-level RE dataset. As we knew, ACE program is to develop the capability to extract meaning from multimedia sources including text, audio and image data. The ACE2004 is designed for text data. There are three types of annotations (entity, relation and event). ACE2004 defines 7 entity types, including person (PER), organization (ORG), geographical entities (GPE), location (loc), facility (FAC), weapon (WEA) and vehicle (VEH). For each pair of entities, it defines 7 relation categories, including physical (PHYS), person-social (PERSOC), employment-organization (EMP-ORG), agent-artifact (ART), PER/ORG affiliation (OTHER-AFF), GPE- affiliation (GPE-AFF) and discourse (DISC). In addition, similar to CoNLL04 dataset, ACE2004 dataset also is widely used on joint entity and relation extraction task.

ACE2005 is still a multilingual corpus for Entity and relation extraction task, which is similar to ACE2004. It is labeled with 7 entity types and 6 relation types. Actually, ACE2005 was built upon ACE2004. Compared with ACE2004, PERSOC, ART and GPE-AFF are kept, but ACE 2005 split PHYS into PHYS and a new relation category PART-WHOLE. It also deleted DISC and merged EMP-ORG and OTHER-AFF into a new category EMP-ORG. The researchers usually use ACE2004 and ACE2005 in the similar way.

SemEval-2010 Task 8 [33] focuses on multi-way classification of semantic relations between pairs of nominals. Compared with CoNLL04, ACE2004 and ACE2005, SemEval-2010 Task 8 has a relatively larger scale. The SemEval-2010 Task 8 dataset contains 10,717 examples annotated with 9 different relation types (*i.e.* cause-effect, instrument-agency, product-producer, content-container, entity-origin, entity-destination, component-whole, member-collection, message-topic) and an additional "other" type, which is used to indicate that the relation in the sentence is covered by the 9 types mentioned above. It is noticeable that the nine relations above are directional, which means the sentence needs to be regarded as expressing a different relationship if the target entities appear in inverse order. For instance, the relation cause-effect(e1, e2) and cause-effect(e2, e1) are considered as two distinct relations. Therefore, many existing works [34, 35] take direction into consideration and the total number of relation types is 19. The SemEval-2010

TABLE 2.2: TACRED statistics per split. About 79.5% of the examples are labeled as no relation (Negative Examples). [2]

Split	Examples	Negative Examples
Training Set	68,124	55,112
Development Set	22,631	17,195
Test Set	15,509	12,184

Task 8 dataset is already partitioned into 8,000 training examples and 2,717 test examples.

TACRED [2] is the most widely used large-scale sentence-level RE dataset published recently. It is a fully supervised dataset of sentence-level binary relation mentions. All 106k examples from TACRED dataset are annotated by crowd workers. The entity mention pairs are collected from the TACKBP4 evaluations 2009–2014, with the years 2009 to 2012 used for training, 2013 for development, and 2014 for testing. Each sentence is labeled with one of 41 person- and organization-oriented relation types, e.g. per:title, org:founded, or the label no relation for negative instances. In addition, TACRED has 42 relation types in total, which are also much more than CoNLL04, ACE2004, ACE2005 and SemEval-2010 Task 8. The key statistics of the TACRED dataset are shown in Table 2.2. We can observe that the distribution of TACRED dataset is a typical long-tail distribution because most samples in this dataset have a no relation label. Therefore, this dataset is closer to the real-world application. Since it is a large scale dataset annotated by crowdsourcing, using Amazon Mechanical Turk. Since there are a few mistakes in the annotations, one recent work [27] studied the errors of the existing models on TACRED dataset. They argue that over 50 percent of labels in TACRED dataset are required to be relabeled. Therefore, they release the TACRED-Revisit (TACREV), which is a modified version of TACRED dataset. In TACREV, authors released the modified development set and test set. They also evaluated the popular baseline models on sentence-level relation extraction on TACREV and these models achieved better performance than original TACRED dataset.

2.2.2 Approaches

At the early stage, researchers used handcrafted features through a series of external tools or manually designing kernels [19] to improve the performance of sentence-level RE, but these approaches suffer from the error propagation issue. Recently, deep neural networks have outperformed previous models using handcraft features. The existing neural models in relation extraction mainly include RNN-based models, Graph-based models and BERT based models. We review these three types of models for sentence-level RE below.

2.2.2.1 RNN-Based Models

RNN is designed for modeling sequence, which has been widely used in different tasks, such as speech recognition [36], audio classification [37] and machine translation [38]. Similarly, RNN is applied and investigated on relation extraction as sentence-level relation extraction is a typical NLP. In [39], authors provide a comprehensive comparison between Convolution Neural Network (CNN) and RNN, which shows that RNN-based model outperforms CNN-based model on sentence-level RE with the same setting. Compared with CNN-based models, the RNN-based models can capture the long-distance patterns, so it is a reasonable solution to learn long-span relations. Several modified versions of RNN-based model were proposed in [39], including a max-pooling feature aggregation, a position indicator approach to specific target nominals, and a bi-directional architecture to learn both forward and backward contexts. These tricks are widely adopted in the following RNN-based models. However, it is debatable whether RNN-based models (*e.g.* LSTM) are capable of memorizing long history[40]. To handle long dependency better, Position-aware LSTM (PA-LSTM) [2] is proposed with TACRED dataset. PA-LSTM uses the relative position of subject and object to enhance the word representations. This approach investigated and combined the semantic similarity-based attention and position-based attention in sentence-level RE. This mechanism can be viewed as the process of performing soft selections of individual words independently. However, the relational expression associated with target entities may be in the form of a segment structure, which means that accurately extracting and modeling such segment information can help the model to identify the informative spans. To achieve this goal, SA-LSTM [22] is proposed to provide

a probabilistic framework for calculating the weights of words globally conditioned on the full sentence. All of these RNN-based models are used to capture sequential information for relation extraction, and these models outperforms handcrafted features by a large margin.

2.2.2.2 Graph-Based Models

Previous works have shown that leveraging syntactic features is helpful for various NLP tasks [41, 42]. However, RNN-based model cannot handle the syntactic information well because RNN-based model treats the context of sentence-level RE as a simple text sequence. Therefore, graph-based models are proposed for sentence-level RE. It is noticeable that most graph-based models are still using RNN-based model to obtain word representations. For instance, C-GCN [24] uses a bi-directional LSTM network to generate contextualized representations, and the contextualized representations and dependency tree structures are fed into the Graph Convolution Network (GCN) to predict the relation. Authors also investigate the interpolation strategy to combine PA-LSTM [2] and C-GCN [24], which achieved better performance than either PA-LSTM or C-GCN. SGC [43] argues the existing relation extraction models inherit unnecessary complexity and redundant computation, so authors proposed to use C-SGC instead of C-GCN to model the dependency structure. However, the simple structure also limits the model to represent more complex semantics. How to effectively make use of relevant information while ignoring irrelevant information from the dependency trees remains a challenging research question. To handle this issue, C-AGGCN [25] is designed by incorporating a dense-connection structure [44] in GCN. C-AGGCN can be understood as a soft-pruning approach that automatically learns to selectively attend to the relevant sub-structures useful for the relation extraction task. However, in C-AGGCN, the graph is fully connected, which destroys the structure information from the original dependency tree. LST-AGCN [45] learns to transform the dependency tree into a weighted graph by considering the syntax dependencies of the connected nodes. The structure of the original dependency tree is also persisted in this manner. LST-AGCN operates on the syntax-transport graph directly to distill the final representation which is sufficient for classification. LST-AGCN achieved the best performance among the graph-based models on sentence-level relation extraction. The graph-based models mentioned above all depend on external parser

tools, so these tools will induce error propagation when the syntactic information is used as the input of GCN. There are also a few works focusing on learning a latent structure from text [46–48]. For instance, in [46], all entities in a sentence are placed as nodes in a fully-connected graph structure, and they represent the edges with position-aware contexts around the entity pairs. A walk-based model is then used to represent and predict the relation by the path between two mentioned entities in relation extraction. The proposed model achieved comparable performance with the model using external tools.

2.2.2.3 BERT-Based Models

BERT-based models achieved the best performance on a number of NLP tasks[49–51]. Many works investigated BERT-based models on sentence-level RE and obtained promising results. BERT-LSTM [52] is one of the most simple BERT-based models on sentence-level RE. They simply put a bi-directional LSTM on the top of BERT, but BERT-LSTM outperforms all graph-based models mentioned above due to the semantic representations from BERT. To further improve BERT-based models and applied BERT on relation extraction more reasonably, R-BERT [53] proposed to use special tokens to label the entities and use the token representations of these entities to predict the target relation, so R-BERT not only leverages the pre-trained BERT language model but also incorporates information from the target entities. BERT-LSTM and R-BERT both focus on the method of leveraging the token representations from BERT in a more elegant way. There are also a few papers focusing on improving the pre-trained language model itself. For instance, SpanBERT [11] is designed to better represent and predict spans of text. SpanBERT extends BERT by: (1) masking contiguous random spans, rather than random individual tokens, and (2) training the span boundary representations to predict the entire content of the masked span, without relying on the individual token representations within it. The novel training process of SpanBERT brings significant improvements on several NLP tasks including sentence-level RE. In addition, Some recent works used external training data to improve sentence-level RE [12, 54, 55]. ERNIE [54] is designed to enhance the language representation model by using both large-scale textual corpora and knowledge graphs. The proposed model can take lexical, syntactic, and knowledge information into consideration simultaneously. KnowBERT [12] is proposed to embed multiple knowledge

TABLE 2.3: Performance of the existing models on TACRED.

Class	Model	<i>Precision</i>	<i>Recall</i>	<i>F1</i>
RNN-based	LSTM [39]	65.7	59.9	62.7
	PA-LSTM [2]	65.7	64.5	65.1
	SA-LSTM [22]	68.1	65.7	66.9
	SA-LSTM+D [22]	69.0	66.2	67.6
Graph-based	GCN [24]	69.8	59.0	64.0
	C-GCN [24]	69.9	63.3	66.4
	AGGCN [25]	69.9	60.9	65.1
	C-AGGCN [25]	73.1	60.9	68.2
	LST-AGCN [45]	-	-	68.8
BERT-based w/o external resource	BERT [10]	69.1	63.9	66.4
	BERT-LSTM [52]	73.3	63.1	67.8
	R-BERT [53]	-	-	69.4
	SpanBERT [11]	70.8	70.9	70.8
BERT-based with external resource	ERNIE [54]	69.9	66.0	67.9
	BERTEM+MTB [55]	-	-	71.5
	KnowBERT [12]	71.6	71.4	71.5
	KEPLER [13]	-	-	71.7
	K-ADAPTER [14]	70.1	74.0	72.0

bases into a single large scale language model. In KnowBERT, there are two objectives, entity linker and self-supervised language modeling, which are jointly optimized in an end-to-end fashion. BERTEM+MTB [55] is designed for improving generalization performance. KEPLER [13] follows KnowBERT, which can not only better integrate factual knowledge into pre-trained language models but also produce effective text-enhanced Knowledge embedding. K-ADAPTER [14] used multiple adapters to learn each kind of infused knowledge.

2.2.2.4 State-of-the-Art

We show the performance of the models we mentioned above in Table 2.3. We can first observe that graph-based models (*e.g.* C-AGGCN) outperform RNN-based models (*e.g.* SA-LSTM). BERT-based models achieved the best performance on sentence-level RE. Among the BERT-based models, BERT [10], BERT-LSTM [52], R-BERT [53] and SpanBERT [11] are trained without other resource, but ERNIE [54], BERTEM+MTB [55], KnowBERT [12], KEPLER [13] and K-ADAPTER [14] used external training resource. Therefore, for the models without

using other training data, SpanBERT [11] is the state-of-the-art sentence-level RE model, and K-ADAPTER [14] is the best one when another training resource is incorporated. In this thesis, we focus on improving the structure of relation extraction model, so we do not use any external training resource. In other words, SpanBERT [11] is the benchmark in our setting.

2.3 Dialogue-level relation extraction

Dialogue-level relation extraction is to predict one relation between two mentioned entities given a piece of dialogue and the two entities. The most important difference between dialogue-level RE and sentence-level RE is the input context. It is a piece of dialogue instead of a single sentence in dialogue-level RE. In addition, in sentence-level RE, each sentence only corresponds to one relation, but each dialogue corresponds to multiple relations in dialogue-level RE. Formally, let $\mathcal{R}$ be the set of predefined relation types. There could be multiple relations $R = \{r_1, r_2, \dots, r_n\}$ between n pairs of entities mentioned in X . The i^{th} relation $r_i \in \mathcal{R}$ is predicted based on an argument pair: subject entity E_s^i and object entity E_o^i . For each prediction, we only provide one pair of mentioned entities and the dialogue X .

2.3.1 Datasets

Dialogue-level RE is a relatively new task comparing with sentence-level RE, so there are less datasets available. In this thesis, we review three datasets, DialogRE [1], MPDD [56] and DDERel [57], respectively.

DialogRE is the first human-annotated dialogue-level RE dataset [1]. The text of DialogRE is collected from the transcripts of American TV situation comedy, Friends. It contains 1,788 dialogues and 36 predefined relation types. An example of this dataset has been given in Table 1.1. Recently, [1] releases a modified English version and a Chinese version of DialogRE. The modified English version fixed a few wrong annotations, and the Chinese version has the same content but different language with the two English versions.

MPDD [56] is a publicly available and the first Chinese dialogue dataset with emotion and relation labels. The conversations are collected from TV series scripts.

Two types of annotations, emotion and interpersonal relationship are labeled. This dataset contains 25,548 utterances from 4,142 dialogues in total. The dialogues include three types of labels - emotion, relation and targeted listener labels. By observing the plot of all the scripts, the relations are divided into 24 types. These relations can also be classified from two perspectives, the social field and the seniority. In the classes of the social field, there are 4 classes, including family, company, school, and others. In the classes of the seniority, interpersonal relation types are separated into 3 classes, *i.e.*, elder, peer, and junior.

DDRel [57] is a very recent dialogue-level relation extraction dataset. The setting of this dataset is understanding interpersonal relationships from dialogue content. Different from DialogRE [1] and MPDD [56], DDRel dataset focuses on the relationship between speakers within different dialogue sessions. All entities in DDRel are speakers. They crawled movie scripts from IMSDb and annotated the relation labels for each session according to 13 pre-defined relationships. The annotated dataset DDRel consists of 6300 dyadic dialogue sessions between 694 pair of speakers with 53,126 utterances in total. Therefore, the models trained by this dataset would be good at generalized to predict interpersonal relationships.

2.3.2 Approaches

In dialogue-level RE, researchers are using different technologies with the sentence-level RE due to the different settings, although some frameworks are similar to each other. In this section, we still review the existing approaches of dialogue-level RE from three aspects, including RNN-based models, graph-based models and BERT-based models.

2.3.2.1 RNN-Based Models

RNN-based models are not good at handling extremely long sequence due to the gradient vanishing and explosion issues. In [1, 57], authors investigated RNN-based models (*e.g.* LSTM [39], BiLSTM [21]) as the baseline models. These RNN-based models use GloVe word embeddings [58], mention embeddings, and type embeddings as input features to predict the relation. Authors assign the same mention embedding to mentions of the same argument and obtain the type embeddings

based on named entity types of the two arguments. spaCy4¹ is used for obtaining entity typing. Among these RNN-based models and even more basic CNN model, BiLSTM [21] is the best one on sentence-level RE. Using Glove embeddings, high-level features are extracted by a single BiLSTM layer. The last hidden states of both directions are concatenated to provide a query for the self-attention among the input words. Finally, the weighted summed feature vector can be used to characterize the whole session and used for final relation classification with a linear layer and a softmax function. In the official implementation, authors use AdamDelta as optimizer with learning rate 0.0003 following [21] on dialogue-level RE. However, the performance of RNN-based models is very limited. One important reason is that the informative words are sparse in a long dialogue, and the memory mechanism of RNN-based models cannot keep long history to model distant dependency, which is more common in dialogue-level RE than sentence-level RE. Therefore, only using RNN-based models cannot model the long distance-dependency in dialogue-level relation extraction well.

2.3.2.2 Graph-Based Models

Graph-based models achieved better performance than RNN-based models on dialogue-level RE. Compared with RNN-based models, graph-based models can handle the long-distance dependency better [59]. [60] implemented several popular graph-based models of sentence-level RE on dialogue-level RE. These graph-based models apply the same word embedding to represent the context, but different graph structures are employed. In [60], they first adopted AGGCN [25] on dialogue-level RE. As we mentioned above, AGGCN used one sentence and full dependency tree of this sentence as input. However, in dialogue-level RE, there are a number of sentences in a piece of dialogue, so authors use the full dependency tree of each sentence as input to construct one graph for each sentence in [60]. In other words, the input includes a piece of dialogue and n graphs in each dialogue, where n is the number of sentences. It is noticeable that the AGGCN updates the edges of graph during forward propagation by attention, so n graphs are connected together as a fully-connected graph within AGGCN to help learn the inter-sentence communication. In addition, LSR [61] is implemented for dialogue-level RE, but it is designed for document-level RE at first. To capture complex interactions between

¹<https://spacy.io/>

inter-sentence entities, authors propose to reason the cross-sentence relationships by a latent graph for document-level RE. Similar with AGGCN, LSR also used the dependency tree to initialize one graph for each sentence. The difference between AGGCN and LSR is the approach to update the adjacent matrix of the graph. LSR adopts an adaptation of Kirchhoffs Matrix-Tree Theorem [62] to induce the latent dependency structure of each dialogue and then feeds the latent structure to a densely connected graph convolutional network to inference the relations. Both AGGCN and LSR mentioned above are designed for sentence-level RE and employed on dialogue-level relation extraction by [60]. In [60], they also proposed DHGAT, which is the best graph-based model on dialogue-level RE. Authors propose an attention-based heterogeneous graph neural network to tackle the dialogue-level RE by using multi-type features to create the graph and employing graph attention mechanism to propagate contextual information. Specially, they use semantic word embedding initialized by GloVe, part-of-speech (POS) tags and named-entity types as the input. They first utilize Utterance Encoder with BiLSTM layers to obtain word representations and encode sentence-level information. Then, using the output of Utterance Encoder, DHGAT constructs a heterogeneous graph with utterance nodes, entity-type nodes, word nodes, speaker nodes. These nodes are connected by the pre-defined rules in their paper, and the graph attention layers are adopted to update the node representations. After the message propagation in the heterogeneous graph, DHGAT selected the entity nodes and argument nodes to predict the relation. Experiments show that their approach achieved outstanding performance on dialogue-level RE.

2.3.2.3 BERT-Based Models

Similar to sentence-level RE, BERT-based models also achieved the best performance on dialogue-level RE. The first BERT-based model is the classical BERT [10]. For dialogue-level RE, the input format of BERT is slightly different from the sentence-level BERT-based models like SpanBERT, although they all use the [CLS] as the start of sequence and [SEP] token after the context. In SpanBERT, people use [unused] token to mask the entities and indicate the BERT-based model to predict the target relation, but existing work [1] appended the two entities at the end of the input and separated them by [SEP] tokens. Also, as the context is different from sentence-level RE, the text of dialogue-level RE includes more speaker-related

TABLE 2.4: Performance of all models on DialogRE. σ denotes the standard deviation computed from five runs of each model.

Class	Model	Dev set		Test set	
		$F1$ (σ)	$F1c$ (σ)	$F1$ (σ)	$F1c$ (σ)
CNN- or RNN-based	CNN [63]	46.1 (0.7)	43.7 (0.5)	48.0 (1.5)	45.0 (1.4)
	LSTM [64]	46.7 (1.1)	44.2 (0.8)	47.4 (0.6)	44.9 (0.7)
	BiLSTM [65]	48.1 (1.0)	44.3 (1.3)	48.6 (1.0)	45.0 (1.3)
Graph-based	AGGCN [25]	46.6	40.5	46.2	39.5
	LSR [61]	44.5	-	44.4	-
	DHGAT [60]	57.7	52.7	56.1	50.7
BERT-based	BERT [10]	60.6 (1.2)	55.4 (0.9)	58.5 (2.0)	53.2 (1.6)
	BERTs [1]	63.0 (1.5)	57.3 (1.2)	61.2 (0.9)	55.4 (0.9)

information. For instance, in dialogue-level RE, the input of BERT can be *[CLS] Speaker 1: Nice to meet you, Tom! Speaker 2: Nice to meet you, too. [SEP] Speaker 1 [SEP] Tom [SEP]*. In dialogue-level RE, the token representation corresponding to the [CLS] token is used as the input of classifier. In BERTs [1], to model the speaker-related information and leverage the overfitting, authors masked the speaker to construct a novel input format for dialogue-level RE. The example we mentioned above would become *[CLS] [unused1]: Nice to meet you! [unused2]: Nice to meet you, too. [SEP] [unused1] [SEP] Tom [SEP]*. The experiments show that the speaker-related information from BERTs can bring significant improvement for dialogue-level RE.

2.3.2.4 State-of-the-Art

We show the performance of the dialogue-level relation extraction models we mentioned above in Table 2.4. The first interesting finding is that AGGCN [25] and LSR [61], two popular graph-based models on sentence-level RE and document-level RE perform weak on dialogue-level RE. Compared with RNN-based models and even CNN-based models, it is interesting that AGGCN and LSR are weak in handling dialogue. One important reason is that both AGGCN and LSR use the syntactic information within a sentence scope, so AGGCN and LSR ignore the inter-sentence communication. However, this kind of information exchange is critically important for dialog-style text, because logical connections are not locally

compact within adjacent sentences. Instead, they are spread over the whole conversations. Therefore, the inter-sentence communication of dialogue provides more information for relational reasoning. On the opposite, DHGAT [60] constructs a heterogeneous graph with shorter distances between logically closed but syntactically faraway word pairs. Hence the long-distance issue is mitigated. We can easily observe that DHGAT outperforms all graph-based models and rnn-based models by a significant margin. However, DHGAT uses BiLSTM to obtain word representations so that it is weaker to model semantic information comparing with BERT-based models. In Table 2.4 BERT [10] and BERTs [1] both outperform DHGAT obviously. BERTs is the best model in Table 2.4. It shows that speaker-related information is helpful to dialogue-level RE.

2.4 Challenges

Although the existing deep learning approaches bring a breakthrough for relation extraction as we mentioned above, there are still a few challenges left. We analyze the potential issues of the existing relation extraction models in this section. First, graph-based models are good at modeling structure information, which is beneficial to both sentence-level relation extraction and dialogue-level relation extraction. Such information is missing in the BERT-based models, although BERT-based models perform well on modeling semantics. Incorporating structure information in BERT-based model would be helpful. Second, graph-based model always relies on the external parser to construct a graph for learning and prediction as input. The external parser induces error propagation, which means the error from dependency tree or other syntactic representations would limit the performance of graph-based models. Third, relation extraction is different from other NLP problems because the target relation we want to predict is weakly related to the semantics of the whole context. However, BERT-based model is designed for modeling that. The performance of BERT-based model would be further improved if we can solve this issue although BERT-based models have achieved state-of-the-art performance on relation extraction. Finally, BERT-based models predict relation by [CLS] token representation designed for modeling the sentence-level semantics. The whole text sequence covers a number of words or tokens unrelated to relation extraction. These words and tokens are noisy for the prediction. We need to use

more refined representations to model the relation. To handle these problems, this thesis proposes GDPNet, which would be introduced in the next chapter.

2.5 Summary

As a key module in many natural language processing applications, relation extraction has received considerable research interest recently. This chapter provided an up-to-date literature survey of sentence-level relation extraction and dialogue-level relation extraction. We first review the existing methods and then summarized their performance. Although deep neural networks bring a breakthrough for relation extraction, there are still a few challenges limiting the models. In this thesis, we devote to solve these issues in the following chapters.

Chapter 3

Approach

In this chapter¹, we propose Gaussian Dynamic Time Warping Pooling Net (GDPNet) to refine latent multi-view graph for relation extraction. Specially, we first claim the problem we want to solve in Section 3.1. The motivation of GDPNet is from the observation of data. To model the asymmetrical relation from a longer sequence, we summarized the potential issues. Then, we introduce our solution in Section 3.2. Specially, we formulize sentence-level relation extraction and dialogue-level relation extraction and introduce the preliminary section in Section 3.2.1. After that, in Section 3.2.2, we propose our model, GDPNet, with Gaussian Graph Generator (GGG) and Dynamic Time Warping Pooling (DTWPool) for relation extraction, which are two main technical contributions of this thesis.

3.1 Potential Problem

We summarized the potential problem we plan to solve in this thesis from three aspects, including graph-based model, BERT-based model and observation from data:

- **Graph-based model.** In relation extraction, graph-based models are usually used to learn syntactic information and long-distance dependency by

¹The work in this chapter has been published as **Fuzhao Xue**, Aixin Sun, Hao Zhang, Eng-Siong Chng "GDPNet: Refining Refining Latent Multi-View Graph for Relation Extraction" Association for the Advancement of Artificial Intelligence (AAAI) 2021

the pre-defined structure of graph. Existing graph-based models (*e.g.* AG-GCN [25]) employ dependency tree from external parser as the graph. These external tools would induce error propagation when these tools provide instruction for graph-based models.

- **BERT-based model.** Relation is predicted by the given context and two entities in both sentence-level RE and dialogue-level RE. BERT-based models use [CLS] token representation of each text sequence to predict the relation, but [CLS] token is designed for covering the semantic information of the whole sequence. The structure information is ignored when we are aiming at classifying an asymmetrical relation.
- **Observation from data.** As we stated in Chapter 1, the indicative words are sparse when the context is long. Both graph-based models and BERT-based models cannot find these indicative words well from a number of words that are unrelated to the target relation.

3.2 Proposed Solution

To solve the issues above, we propose a unified model, GDPNet, which can be applied on both sentence-level RE and dialogue-level RE. We will introduce the proposed model in this section.

3.2.1 Preliminary

Before introducing our GDPNet in detail, we first formulate sentence-level RE and dialogue-level RE. Two important preliminary modules, multi-view graph and SoftDTW would also be introduced before GDPNet.

3.2.1.1 Problem Formulation

Let $X = \{x_1, x_2, \dots, x_T\}$ be a sequence, where x_t is the t^{th} token in the sequence, and T is the number of tokens. For sentence-level RE, X denotes the given sentence. For dialogue-level RE, X represents the entire dialogue. That is, in our problem

formulation, we do not explicitly distinguish sentence and dialogue. To predict relations, we are given two entities, subject entity X_s and object entity X_o . Both X_s and X_o are sub-sequences of X . An entity may contain one or more tokens, *e.g.*, $X_s = \{x_s, x_{s+1}, \dots, x_{s+m-1}\}$ where s denotes the starting position of X_s in sequence X and m is the number of tokens in X_s . Given X , X_s , and X_o , the goal of relation extraction is to predict the relation $r \in R$ between X_s and X_o , where R is a set of predefined relation types in the dataset.

3.2.1.2 Multi-View Graph

FIGURE 3.1: Comparison between homogeneous graph and multi-view graph

In a homogeneous graph, there exist one or no edge between a pair of nodes. Formally, we can represent a homogeneous graph as $G = (V, A)$, where V is a set of nodes, A is an adjacent matrix.

However, as shown in Figure 3.1, in a multi-view graph, there may exist multiple edges between a pair of nodes, each edge from one view. Formally, we can represent a multi-view graph as $G = (V, A^1, A^2, \dots, A^N)$, where V is still a set of nodes, A^n is n_{th} adjacent matrix, and N is the number of views.

In our model, we aim to use multi-view graph to model the complex (*e.g.*, syntactic and semantic) relationships between tokens in the given sequence. In this multi-view graph, each token in sequence X corresponds to one node. Note that an entity may contain multiple tokens, and there are multiple ways of handling entity tokens (*i.e.*, tokens in X_s and X_o). For easy presentation, we treat tokens in entities the same as other tokens in the sequence in the following discussions.

3.2.1.3 SoftDTW

Dynamic time warping (DTW) [66, 67] is a modified dynamic programming algorithm designed for speech recognition. SoftDTW is a loss function proposed for DTW discrepancy and global alignment kernel (GAK) [68]. Both DTW and GAK can be used to compare two time series $\mathbf{x} = (x_1, x_2, \dots, x_m) \in \mathbb{R}^{p \times m}$ and $\mathbf{y} = (y_1, y_2, \dots, y_n) \in \mathbb{R}^{p \times n}$, where the length of $\mathbf{x}$ is m and the length of $\mathbf{y}$ is n . There is no constraint on the value of m and n .

Dynamic time warping. We first consider a differentiable substitution-cost function:

$$\delta : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}_+ \quad (3.1)$$

This function is then used to define a cost matrix:

$$\Delta(\mathbf{x}, \mathbf{y}) := [\delta(x_i, y_j)]_{ij} \in \mathbb{R}^{n \times m} \quad (3.2)$$

Given an alignment matrix A in $\mathcal{A}_{ij}$, where $\mathcal{A}_{n,m} \subset \{0, 1\}^{n \times m}$ is the set of binary alignment matrices, DTW is defined as:

$$\text{DTW}(\mathbf{x}, \mathbf{y}) := \min_{A \in \mathcal{A}_{n,m}} \langle A, \Delta(\mathbf{x}, \mathbf{y}) \rangle \quad (3.3)$$

Global alignment kernel. Similar to DTW, GAK also considers the costs of all possible alignment matrices, but the way to compute is different. GAK is defined as:

$$k_{\text{GA}}^\gamma(\mathbf{x}, \mathbf{y}) := \sum_{A \in \mathcal{A}_{n,m}} e^{-\Delta(\mathbf{x}, \mathbf{y})/\gamma} \quad (3.4)$$

We can observe that GAK uses a summation operator instead of a minimum to calculate the cost.

In SoftDTW [26], authors proposed that both formulas in Eq.3.3 and Eq.3.4 can be computed with a single algorithm. To achieve this goal, they first proposed a generalized min operator as following:

Algorithm 1: Forward recursion to compute $\mathbf{dtw}^\gamma$ and intermediate alignment costs

Input: $\mathbf{x}, \mathbf{y}$, smoothing $\gamma \geq 0$, distance function δ

$r_{0,0} = 0; r_{i,0} = r_{0,j} = \infty$

1: **for** $j=1,2,\dots,m$ **do**

2: **for** $i=1,2,\dots,n$ **do**

3: $r_{i,j} = \delta(x_i, y_j) + \min^\gamma \{r_{i-1,j-1}, r_{i-1,j}, r_{i,j-1}\}$

4: **end for**

5: **end for**

Output: $(r_{m,n}, R)$

$$\min^\gamma \{a_1, a_2, \dots, a_n\} := \begin{cases} \min_{i \leq n} a_i, & \gamma = 0 \\ -\gamma \log \sum_{i=1}^n e^{-a_i/\gamma}, & \gamma > 0 \end{cases} \quad (3.5)$$

where γ is a smoothing parameter and $\gamma \geq 0$. We can define SoftDTW loss with the operator defined at Eq.3.5:

$$\mathbf{dtw}^\gamma(\mathbf{x}, \mathbf{y}) := \min^\gamma \{ \langle A, \Delta(\mathbf{x}, \mathbf{y}) \rangle, A \in \mathcal{A}_{n,m} \} \quad (3.6)$$

In SoftDTW (Eq 3.6), we still consider the costs of all possible alignment matrices as DTW and GAK. When γ is setting to 0, the original DTW score is recovered. When $\gamma \geq 0$, we can obtain the following equation:

$$\mathbf{dtw}^\gamma = -\gamma \log k_{\text{GA}}^\gamma \quad (3.7)$$

where k_{GA}^γ is the GAK we defined in Eq. 3.4. Therefore, both DTW and GAK are incorporated in SoftDTW. Most importantly, and in either case, $\mathbf{dtw}^\gamma$ can be computed by Algorithm. 1, so both operations cost and storage cost are (nm) . Also, please note that the operator $\min^\gamma$ must be computed using the usual log-sum-exp stabilization trick to ensure numerical stability, which can be written as:

$$\log \sum_i e^{z_i} = (\max_j z_j) + \log \sum_i e^{z_i - \max_j z_j} \quad (3.8)$$

In this thesis, we use SoftDTW to improve graph pooling and propose DTWPool. We will introduce DTWPool in the following parts.

FIGURE 3.2: The overall architecture of the proposed GDPNet. Entities E_1 and E_2 are single-token entities in the illustration.

3.2.2 GDPNet

GDPNet is the main model we proposed in this thesis. There is a BERT module and a graph module incorporated, so GDPNet is either a BERT-based model or a graph-based model because we put a graph module on top of BERT module. In graph module, we propose Gaussian Graph Generator (GGG) and Dynamic Time Warping Pooling (DTWPool), which are two main contributions of this thesis.

We first introduce GDPNet in general. The overall architecture of GDPNet is shown in Figure 3.2. There are three key components: BERT module, graph module, and SoftMax classifier. The BERT module encodes tokens X into the corresponding feature representations H . The graph module takes in token representations from BERT except for the sequence token representation h_0 . These token representations construct a refined task-specific graph V as the output of graph module. The final task-specific graph V and sequence representation h_0 are fed into the SoftMax classifier to predict relation type.

3.2.2.1 BERT Module

We utilize BERT-based model as the feature encoder to extract token representations due to its effectiveness in representation learning [1, 11]. The BERT-based model can be different when we are working on different relation extraction tasks. In this thesis, we use SpanBERT [11] as feature encoder for sentence-level RE, and BERTs [1] is used for dialogue-level RE. Therefore, we introduce BERTs and SpanBERT in this section. Since they are both based on the classical BERT [10], we introduce the classical BERT before BERTs and SpanBERT.

BERT [10] is a well-known language model in natural language processing. It is a self-supervised learning model designed for pre-training a deep transformer encoder before fine-tuning it for a specific downstream NLP task. Formally, the input of BERT is denoted as a sequence of words or sub-word tokens $X = (x_1, x_2, \dots, x_n)$, and BERT would encode a sequence to context representations $H = (h_1, h_2, \dots, h_n)$. Each token representation corresponds to one token in the input text. Instead of using traditional left-to-right or right-to-left language models [69, 70], BERT used masked language model (MLM) and next sentence prediction (NSP) during pre-training:

- MLM is to predict the replaced tokens in the input context so it can help model to learn token-level representations. About 15 percent of tokens $Y \subseteq X$ are randomly selected. These selected tokens are divided into three different groups, each group has a unique way to deal with the tokens. Among them, the first way is that 80 percent of them are masked by the special tokens [MASK] pre-defined by BERT, and the second way is that 10 percent of the selected tokens are replaced by a random token, which is sampled from a unigram distribution. The last one is to keep 10 percent of the tokens unchanged. The goal of MLM is to recover the original input text.
- NSP is designed to learn sentence-level representations by self-supervised learning from text without annotation. The input of this task is a pair of sentences (X_A, X_B) , and the target is to predict whether X_B is the direct continuation of X_A . The result of prediction depends on the [CLS] token which locates at the start of BERT input, which means this training strategy

FIGURE 3.3: Input format of BERT and BERTs

encourages the [CLS] token representation to cover the sentence-level semantic information. The two sentences X_A and X_B are separated by the [SEP] token.

BERTs is a speaker-aware modification of BERT [10], which achieved the best performance on dialogue-level RE. It has the same architecture with BERT and uses pre-trained BERT as backbone to fine-tune the parameters for dialogue-level RE. The difference between BERTs and BERT is the input format. To present the idea of BERTs, we first formulate one dialogue as $D = s_1 : t_1, s_2 : t_2, \dots, s_m : t_m$, where s_i and t_i denote the speaker ID and text of the i^{th} turn, respectively, and m is the total number of turns in a dialogue. (E_s, E_o) is a pair of entities provided for predicting the relation. The visualization of the input of BERT and BERTs are shown in Fig 3.3. The input sequence of BERT is $\langle [\text{CLS}], D, [\text{SEP}], E_s, [\text{SEP}], E_o, [\text{SEP}] \rangle$. Please note that either E_s or E_o may contain multiple tokens. Inspired by the data format of dialogue, authors proposed a new input sequence with two motivations: (1) helping the model locate the start positions of relevant turns based on the arguments that are speaker names; (2) preventing the model from overfitting to the training data. Specially, we construct a new dialogue $\hat{D} = \hat{s}_1 : t_1, \hat{s}_2 : t_2, \dots, \hat{s}_m : t_m$, where $\hat{s}_i$ is defined as:

$$\hat{s}_i = \begin{cases} [S_1], & \text{if } s_i = E_s \\ [S_2], & \text{if } s_i = E_o \\ s_i, & \text{otherwise} \end{cases} \quad (3.9)$$

where $[S_1]$ and $[S_2]$ are two special tokens designed for BERTs here. At the same time, we replace E_s or E_o by $[S_1]$ or $[S_2]$, respectively, if E_s or E_o is a speaker in the dialogue. We named E_s or E_o as $\hat{E}_s$ or $\hat{E}_o$ no matter whether they are replaced by $[S_1]$ or $[S_2]$. Therefore, the input sequence of BERTs is $\langle [\text{CLS}], \hat{D}, [\text{SEP}], \hat{E}_s, [\text{SEP}], \hat{E}_o, [\text{SEP}] \rangle$.

SpanBERT [11] is different from BERTs designing for a specific relation extraction task. Instead, SpanBERT is proposed as a modified version of BERT, which means SpanBERT is a language model instead of a relation extraction model. The goal of SpanBERT is to improve pre-training of BERT by representing and predicting spans of text. As an extension of BERT, for SpanBERT, there are two main differences from BERT, the masking scheme and the training objectives. In BERT, authors mask the tokens of input text sequence randomly, but SpanBERT considers the span of the tokens instead of treating the tokens independently. Specially, SpanBERT samples a span length (number of words) l from a geometric distribution $Geo(p)$, so the shorter span is preferred during sampling, where p is a hyper-parameter set as 0.2 through preliminary experiments. The starting point of the span used to mask is randomly selected according to a uniform distribution. To avoid the overlength span, authors clip l at $l_{max} = 10$. The mean span length of l is 3.8 during pre-training. Since there are different types of tokens selected from raw text, SpanBERT replaces all the tokens in a span with [MASK] token or randomly sampled tokens. On sentence-level RE, SpanBERT is simply fine-tuned based on the classification of relation. It uses [CLS] token representation to predict the target relation.

In the BERT module of this thesis, we use BERTs for dialogue-level RE and SpanBERT for sentence-level RE. Either BERTs or SpanBERT is used to encode the context into token representations. For dialogue-level RE, we use the same input format $\langle [\text{CLS}], \hat{D}, [\text{SEP}], \hat{E}_s, [\text{SEP}], \hat{E}_o, [\text{SEP}] \rangle$ with BERTs. For sentence-level RE, the pre-trained SpanBERT is employed as the backbone, and we train our GDPNet by fine-tuning it. Since either BERTs or GDPNet works as a context encoder, we can obtain $H = \{h_0, h_1, h_2, \dots, h_T\}$ as token representations from input text $X_{input} = \{x_0, x_1, x_2, \dots, x_T\}$. Existing BERT-based solutions for both sentence-level RE and dialogue-level RE only take h_0 , the representation of [CLS] token as the input of classifier. In GDPNet, we fully utilize the entire token representations H through the graph module. To be detailed shortly, the graph module learns a task-specific graph using tokens $\{h_1, h_2, \dots, h_T\}$. The learned graph is then combined with h_0 to be the input to the final classifier.

FIGURE 3.4: The overall architecture of the graph module.

3.2.2.2 Graph Module

Graph module is on top of BERT module as we shown in Figure. 3.2. It takes token representations learned $\{h_1, h_2, \dots, h_T\}$ from BERT module as input.² The main contributions of this thesis, Gaussian Graph Generator (GGG) and Dynamic Time Warping Pooling (DTWPool) are focused on the graph module in GDPNet. To introduce graph module clearly, an overview of the graph module of GDPNet is drawn in Figure. 3.4. We can observe that the token representations are served as nodes in the graph, and these nodes are fed into the Gaussian Graph Generator to generate a latent multi-view graph. Then, this graph is updated by graph convolution layers and DTWPool layers iteratively. Therefore, we will obtain a smaller task-specific graph with much fewer nodes. This task-specific graph is concatenated with the original graph with dense connection [71] to obtain the final graph, which would be used to predict the relation with [CLS] token representation (*i.e.*, h_0) learned from BERT module. In this part, we follow the order of data flow to introduce each component of the graph module in our GDPNet.

Gaussian Graph Generator is designed to generate a latent multi-view graph for relation extraction. In BERT module, we obtain the token representations $H = \{h_0, h_1, h_2, \dots, h_T\}$, and these token representations are divided into two parts in the graph module. The first part is the [CLS] token representation h_0 , which is fed into the classifier directly. The second part is $\{h_1, h_2, \dots, h_T\}$, which incorporates all token representations except h_0 . Since we are describing a graph module

²Please note that h_0 is not used here.

in this part, we use the original node representations $V^0 = \{v_1^0, v_2^0, \dots, v_T^0\}$ to denote the input token representations. The graph module of GDPNet is designed to model the asymmetrical relation from a longer sequence by a task-specific graph. Therefore, we need to generate edges according to the node representations from BERT module. Since our graph is to model the semantic relation, the structure of the homogeneous graph is too simple to model the complex semantic relation. Therefore, in this work, we propose to use the multi-view graph introduced in Section. 3.2.1.2 to model the semantic relation in the given context. We propose Gaussian Graph Generator (GGG) to initialize the edges of the latent multi-view graph, based on V^0 . Specifically, we first encode each node v_i^0 into multiple Gaussian distributions. We first generate the mean as:

$$\{\mu_i^1, \mu_i^2, \dots, \mu_i^N\} = g_\theta(v_i^0) \quad (3.10)$$

where g_θ is a trainable neural network, and N denotes the number of views in the multi-view graph. We encode each node to N mean values of Gaussian distributions. We then encode the standard deviation of Gaussian distributions:

$$\{\sigma_i^1, \sigma_i^2, \dots, \sigma_i^N\} = \phi(g'_\theta(v_i^0)) \quad (3.11)$$

where g'_θ is a trainable neural network, ϕ is a non-linear activation function. We set the activation function ϕ as the SoftPlus function here, since the standard deviation of Gaussian distribution is bounded on $(0, +\infty)$. Consequently, we obtain a number of Gaussian distributions $\{\mathcal{N}_1^n, \mathcal{N}_2^n, \dots, \mathcal{N}_{T+1}^n\}$ for the n^{th} view of the multi-view graph. Each Gaussian distribution here $\mathcal{N}_i^n(\mu_i^n, \sigma_i^{n2})$ corresponds to one node representation v_i^0 .

In GDPNet, the purpose of the multi-view graph is to capture all possible semantic relations between tokens, and we obtained the Gaussian distributions from the corresponding node representations. We then move to construct the edges by the Gaussian distributions learned. Since the informative tokens are sparse in the long text sequence, we encourage message propagation between token representations with large semantic differences. We adopt KL divergence between the Gaussian distributions of two tokens to model edge weight. Specifically, edge weight between

i^{th} node and j^{th} node on the n^{th} view is computed as:

$$e_{ij}^n = \text{KL}(\mathcal{N}_i^n(\mu_i^n, \sigma_i^{n2}) || \mathcal{N}_j^n(\mu_j^n, \sigma_j^{n2})) \quad (3.12)$$

After computing edges between nodes on each view, we obtain multiple adjacent matrices $\{A^1, A^2, \dots, A^N\}$, one for each view. Thus, the generated multi-view graph is written as $G = (V^0, A^1, A^2, \dots, A^N)$. Due to the asymmetry nature of KL divergence, G is a directed multi-view graph. The weight of e_{ij}^n is bounded on $[0, +\infty)$. When the Gaussian distributions are the same, $e_{ij}^n = 0$. In other words, there is no edge for the same node, and the main diagonal of the adjacency matrix equals zero.

Multi-view graph convolution is used to update the graph iteratively in GDPNet. Given the graph we generated above, our next goal is to update the graph and learn relations from it. Since the latent graph in GDPNet is a multi-view graph instead of a homogeneous graph, we are required to employ a graph convolution algorithm that can be used to update a multi-view graph. To this end, we follow the Densely Connected Graph Convolutional Networks (DCGCN) [44] to model the asymmetrical relation. DCGCN was first proposed for graph-to-sequence learning. The authors proposed to use DCGCN to integrate both local and non-local features to learn a better structural representation of a graph, and they evaluated the performance on the text generation task. AGGCN [25] also referred to DCGCN to apply the graph structure learned to improve relation extraction. In AGGCN, the initialization of the graph, a dependency tree [72], is given by an external parser. Each node of the graph corresponds one word embedding. However, in our GDPNet, to make use of better semantic representations from BERT-based model, we use token representation learned from BERT module as node. In addition, we can also avoid error propagation from external parser tool. Dense connectivity is an important component in DCGCN. The output of the current layer is directly connected to all its preceding layers. To introduce the multi-view graph convolution in GDPNet, we first define $k_i^{(\ell)}$, which is the concatenation of the initial i^{th} node representation and i^{th} node representations generated by the preceding layers $1, \dots, \ell - 1$:

$$k_i^{(\ell)} = [v_i^0, k_i^{(1)}, \dots, k_i^{(\ell-2)}, k_i^{(\ell-1)}] \quad (3.13)$$

where v_i^0 is the original node representations of i^{th} node in our latent multi-view graph. We can see that $k_i^{(\ell)}$ is densely connected to the output of previous layers so it can help us to stack more graph convolution layers without oversmoothing problems.

With the usage of dense connections, we can train a deeper model to capture both local and non-local information. The multi-view graph convolution is written as:

$$v_{n_i}^{(\ell)} = \rho \left(\sum_{j=1}^T A_{ij}^n W_n^{(\ell)} k_j^{(\ell)} + b_n^{(\ell)} \right) \quad (3.14)$$

where $W_n^{(\ell)}$ and $b_n^{(\ell)}$ are the trainable weight and bias of the n^{th} view, respectively. ρ denotes an activation function, and we use ReLU [73] function in this thesis. $k_j^{(\ell)}$ is the concatenation of the initial node representation and the node representations produced in sub-layers $1, \dots, \ell - 1$. Please note that we use different $W_n^{(\ell)}$ and $b_n^{(\ell)}$ when we are computing the different output of sub-layers. For instance, we use $W_n^{(i)}$ and $b_n^{(i)}$ when we are aiming at updating the node representations in the i^{th} sub-layer. The n^{th} view output of the first graph convolution layer is:

$$V_n^1 = \{v_{n_1}^1, v_{n_2}^1, \dots, v_{n_{T+1}}^1\} = \{k_{n_1}^{(\ell)}, k_{n_2}^{(\ell)}, \dots, k_{n_{T+1}}^{(\ell)}\} \quad (3.15)$$

Since we have multiple views in the graph, we are interested in combining the node representations learned from different views. Finally, the graph after the first graph convolution layer in the graph module can be denoted as:

$$V^1 = \{V_1^1, V_2^1, \dots, V_N^1\} \quad (3.16)$$

After node representations, the A^n adjacent matrix from n^{th} view is updated by attention mechanism:

$$A^n = softmax\left(\frac{QW_n^Q \times (KW_n^K)^T}{\sqrt{d}}\right) \quad (3.17)$$

where Q and K are query and key, respectively, and they both equal to the node representations in V^1 . W_n^Q and W_n^K are two trainable parameters of n^{th} head, which can be used to map the node representations to an embedding. We calculate

the weight of the edges in n^{th} view of the latent multi-view graph, so the n^{th} adjacent matrix A^n can be seen as the output of n^{th} attention head.

Dynamic Time Warping Pooling is introduced to refine the latent multi-view graph after graph convolution that updates node representations by message propagation. We first introduce Self-Attention Graph Pooling (SAGPool) [74] because our DTWPool is a modified version of SAGPool.

SAGPool is a graph pooling method using graph convolution to consider both node representations and graph topology in graph pooling. Since self-attention [75] is designed to use input features as the criteria for the attention itself, authors propose to use the combination of self-attention and graph convolution to learn the self-attention scores. When we use the graph convolution formula in [76], the self-attention score $Z \in \mathbb{R}^{N \times 1}$ can be denoted as:

$$Z = \sigma(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{\frac{1}{2}} X \Theta_{att}) \quad (3.18)$$

where σ is an activation function, and we set it as *tanh* in this work. $\hat{A}$ equals to $A + I_N$, which means $\hat{A}$ is an adjacency matrix with self-connections. $\hat{D} \in \mathbb{R}^{N \times N}$ is the degree matrix of $\hat{A}$. $X \in \mathbb{R}^{N \times F}$ denotes node representations of the input graph here, and Θ_{att} is a trainable matrix, which can be optimized by backpropagation algorithm in an end-to-end fashion. Please note Θ_{att} is the only trainable parameter of the SAGPool layer. Since authors used graph convolution operators to learn the self-attention score, the scores provided by model take both node representations and topology information into consideration. Then, according to the self-attention score Z , we can use a node selection algorithm to obtain a refined graph. SAGPool follows Graph u-net [77] to select the nodes by preserving the fixed portion of nodes of the input graph. In other words, different from many existing graph pooling methods [78], the top-k approach [77] used can refine the graph with different scales and structure. In top-k approach, one hyperparameter $k \in (0, 1]$ denotes the portion of nodes to keep during the graph pooling, which means $\lceil kN \rceil$ nodes can be retained in SAGPool. Specially and formally, the id of selected nodes can be written as:

$$\text{idx} = \text{toprank}(Z, \lceil kN \rceil) \quad (3.19)$$

where Z represents the attention scores generated above, and then idx can be calculated based on the number of nodes $\lceil kN \rceil$ we are interested in. `toprank` is a ranking function to obtain the indices of the top values. In the implementation, we simply modify the attention mask matrix to control the nodes considered:

$$Z_{mask} = Z_{idx} \quad (3.20)$$

where Z_{idx} represents the nodes selected. For the nodes with a high attention score, the corresponding index equals 1 in Z_{idx} , and we update the attention mask matrix Z_{mask} to consider the structure refined.

To obtain the refined graph, SAGPool updates the node representations X to X_{out} by the following equation:

$$X_{out} = X_{idx,:} \odot Z_{mask} \quad (3.21)$$

where $X_{idx,:} \in \mathbb{R}^{\lceil kN \rceil \times F}$ denotes node representations selected from X by idx . $\odot$ is the broadcasted elementwise product.

Then, the adjacent matrix is also required to be updated to match the scale of nodes:

$$A_{out} = A_{idx,idx} \quad (3.22)$$

where $A_{idx,idx} \in \mathbb{R}^{\lceil kN \rceil \times \lceil kN \rceil}$ represents the refined adjacent matrix selected from A . Therefore, in summary, for a homogeneous graph $G = (X, A)$, the output of SAGPool is $G_{out} = (X_{out}, A_{out})$

One main advantage of SAGPool is that it can refine the graph based on both the node representations and topology information. However, for our task, SAGPool has a few weakness:

- SAGPool can consider the node representations and topology information, but the latent graph is learned from a text sequence in GDPNet. Therefore, it would be helpful to consider the sequential information when we refine the latent graph.

- SAGPool can refine the graph with different scales by a fixed pooling ratio. However, the portion of informative nodes may be varied in different input graphs, so we need an adaptive graphing pooling method to refine the latent graph in this thesis,
- Input graph of SAGPool is a homogeneous graph. To model complex semantic relation, however, our goal is to refine a latent multi-view graph learned from text. Expanding the graph pooling methods to multi-view graph is the problem we want to solve in this thesis.

To solve the issues mentioned above, we propose the DTWPool as a modified graph pooling method based on SAGPool. DTWPool is designed for the latent multi-view graph learned from sequence. We refer to SAGPool to calculate the attention scores on each view of the graph:

$$s_{n_i} = \alpha \left(\sum_{j=1}^T A_{ij}^n W_{pool} v_j + b_{pool} \right) \quad (3.23)$$

where W_{pool} and b_{pool} are trainable weight and bias of the pooling operation, respectively. α denotes an activation function and s_{n_i} is the attention weight before the SoftMax activation. For n^{th} view of the latent multi-view graph, we obtain a score set $S_n = \{s_{n_1}, s_{n_2}, \dots, s_{n_{T+1}}\}$. We keep the node selection method of SAGPool to retain a portion of nodes in the input graph even when the sizes and structures of the graphs are varied. After node selection, the retained nodes of n^{th} view are a subset of the V^1 , *e.g.*, $V_n^2 = \{v_1^2, v_5^2, \dots, v_T^2\}$. As our latent graph has multiple views, we can derive different subsets from V^1 from different views.

Existing graph pooling approaches, *e.g.*, SAGPool, only allow a fixed ratio for node pooling. Due to the nature of multi-view graph, DTWPool can refine the graph adaptively by getting the union set of nodes from different views:

$$V^2 = V_1^2 \cup V_2^2 \cup \dots \cup V_N^2 \quad (3.24)$$

where V^2 is the union set of the subsets selected from all different views. If we set a fixed pooling ratio $r \in [0, 1]$ on each view, the pooling ratio of DTWPool, *e.g.*, ratio of the number of nodes in V^2 to the number of nodes in V^1 , could be

a flexible decimal $r_{real} \in [r, 1]$. Following SAGPool, our DTWPool updates the adjacent matrix based on the nodes selected V^2 according to Eq 3.22.

We introduced the Gaussian Graph Generator to generate a latent multi-view graph, and then we update and refine the graph by multi-view graph convolution and Dynamic Time Warping Pooling proposed, respectively. To obtain the task-specific graph for relation extraction, we operate the graph convolution and DTWPool iteratively in the graph module of this work, so we have a sequence of graphs $\{G^1, G^2, \dots, G^D\}$, where D is the number of graph pooling layers.

The number of informative nodes varies in different text sequences. It would be helpful to preserve important information along the process of graph pooling. The nodes in this graph embed rich context information, so it would be beneficial to summarize the context into the pooled nodes. To this end, we propose to adopt SoftDTW to guide the graph pooling. The detailed description of SoftDTW can be found in Section 3.2.1.3.

In GDPNet, we use the SoftDTW loss to minimize the distance between the original graph and the last pooled graph:

$$\mathcal{L} = \text{CSE}(r, \hat{r}) + \lambda \text{DTW}_\gamma(V^1, V^D) \quad (3.25)$$

where $\mathcal{L}$ denotes the overall training objective, CSE is the Cross-Entropy loss function, Please note we use Binary Cross-Entropy loss function BCE instead of CSE in dialogue-level relation extraction. λ is a hyper-parameter to balance the contribution of DTW, and V^D is the nodes in the graph after the last DTWPool layer. Because SoftDTW loss is designed for aligning two sequences, we employ it to encourage the nodes after pooling to cover more local context representations. With SoftDTM loss, DTWPool is guided to refine the graph without losing much context information.

To minimize information loss and alleviate the oversmoothing problem from multi-view graph convolution, we concatenate the node representations of the intermediate graphs created during the pooling process to derive the final graph V , similar to learning graph dense connections [44]. Our pooled graphs have different sizes, and we are only interested in the smaller task-specific graph in this thesis. Therefore, we concatenate the node representations in $\{V^2, V^4, \dots, V^D\}$ for all nodes that are included in graph V^D , which means the number of nodes in the final graph V is

the same as that in V^D . V is the task-specific graph we are interested in and the final output of our graph module in GDPNet.

3.2.2.3 Classifier

Given the final graph V , we adopt a neural network with max-pooling to compute the representation of the graph. The computed representation is then concatenated with the representation of [CLS] token h_0 to form the final representation.

$$h_{final} = [h_0; f(V)] \quad (3.26)$$

Here, f is a neural network with max-pooling, which maps the $V \in \mathbb{R}^{Q \times T'}$ to $f(V) \in \mathbb{R}^{q \times 1}$, $Q = D * q$. D is the number of graph pooling layers in graph module, q is the dimension of token representation, and T' is the number of nodes in V .

3.3 Summary

This chapter introduced GDPNet, the model we proposed in this thesis. GDPNet is designed for improving relation extraction by refining the latent multi-view graph. It incorporates BERT module, graph module and a final classifier. The main contributions of our thesis are focused on the graph module. We first encode text to token representations in BERT module and use the token representations as nodes to construct a latent multi-view graph by GGG. To further refine the latent multi-view graph, we propose DTWPool, a graph pooling method to refine the multi-view graph learned from sequence. The graph is updated through multiple interactions of multi-view graph convolution and DTWPool. Finally, the refined latent graph is fed into the classifier to predict relation type.

Chapter 4

Experiments

In this thesis¹, we propose a unified model, GDPNet, which can be applied on both sentence-level and dialogue-level RE tasks. Due to the differences in data formats, applicable baseline models, and the way in handling subject and object entities X_s and X_o , we conduct two sets of experiments in this chapter to compare GDPNets to SoTA models on the two tasks. In Section 4.1, we first evaluate our model on dialogue-level RE, and then we evaluate our GDPNet on sentence-level RE in Section 4.2. Finally, to further justify that our model can identify the inductive tokens to improve relation extraction, we analyze DTWPool by a set of quantitative analysis in Section 4.3.

The GDPNet is implemented by using PyTorch 1.4 with CUDA 10.1. Our implementation also uses the SoftDTW² toolkit. All experiments are conducted on a desktop with Intel i7-8750H CPU, DDR4 16GB memory, and a single NVIDIA GeForce RTX 1070 GPU. We also reproduced our results on Quadro RTX 8000 GPU.

¹The work in this chapter has been published as **Fuzhao Xue**, Aixin Sun, Hao Zhang, Eng-Siong Chng "GDPNet: Refining Refining Latent Multi-View Graph for Relation Extraction" Association for the Advancement of Artificial Intelligence (AAAI) 2021

²<https://github.com/Maghoumi/pytorch-softdtw-cuda>

4.1 Dialogue-level Relation Extraction

4.1.1 Datasets

DialogRE is the first human-annotated dialogue-level RE dataset [1]. It contains 1,788 dialogues originating from the complete transcripts of a famous American television situation comedy. There are 36 relation types predefined in DialogRE. An example has been given in Table 1.1. In each dialogue, there are multiple relations between the entities, and we focus on predicting one relation in one forward process in this work. In other words, given a pair of entities and the context, we are aiming at predicting one relation once although there are multiple pairs of entities given in one dialogue.

4.1.2 Baseline Models

We evaluate GDPNet against the recently proposed BERTs [1]. BERTs is a speaker-aware modification of BERT. As we mentioned above, it achieved the best performance on dialogue-level RE before us. We also include other popular baseline models in dialogue-level RE. For RNN-based models, we include two typical baseline models LSTM [64] and BiLSTM [65] in our experiments. We also incorporate a few graph-based models for the completeness of experiments. DHGAT [60] is the best graph-based model to our best knowledge. AGGCN [25] and LSR [61] have also been employed on dialogue-level RE recently.

4.1.3 Experimental Setup

For a fair comparison, we use the same input format and hyperparameter settings as in BERTs as BERTs is our strongest and direct baseline. Specifically, the given X , X_s , and X_o , are concatenated with classification token [CLS] and separator token [SEP] to form an input sequence [CLS] X [SEP] X_s [SEP] X_o [SEP]. All token representations except [CLS] are fed into our graph module. To incorporate speaker information, for the sentences that contain X_s or X_o , the text indicating speaker *e.g.* “Speaker 1”, is replaced by a specific token, $[S_1]$ or $[S_2]$. Note that the trigger words are treated as normal tokens, which means our model cannot see the

TABLE 4.1: Hyperparameter settings on DialogRE.

Parameter	Value
Epoch	20
Batch Size	24
Learning rate	3e-5
Dropout	0.5
Hidden units of graph	300
Number of views	3
Number of DTWPool layers	3
Pooling ratio lower bound	0.7
Weight of SoftDTW loss	1e-6

special annotations of the trigger words. The hyperparameter settings are shown in Table 4.1. We set hyper-parameters (epoch, batch size, learning rate and dropout rate) of the backbone model the same as BERTs [1] for a fair comparison. The hidden units of graph, number of views, and number of DTWPool layers are set according to AGGCN [25]. Although AGGCN does not contain graph pooling operation, we use the same number of GCN layers and DTWPool layers in our evaluation. We set a smaller pooling ratio lower bound 0.7 for DialogRE because dialogues are usually relatively long. Adam [79] with a learning rate of $3e - 5$ is employed. We use 3 DTWPool layers. As graph pooling operation is performed in each DTWPool layer, only a few nodes are included in the final graph. We use both $F1$ and $F1c$ scores as the evaluation metrics. $F1c$ is proposed by [1], and it is computed by only taking in the early part of dialogue as input, instead of the entire dialogue.

4.1.4 Results on DialogRE

We first summarize the results on DialogRE in Table 4.2. Since the scale of DialogRE is not large, we also report the mean and the standard deviation σ in our experiments by running the code with different random seeds. Please note that the random seeds are randomly selected in this paper. In Table 4.2, we can observe that BERT-based models significantly outperform CNN LSTM-based and BERT-based models. For instance, BERT is the most basic BERT-based model in dialogue-level RE, and DHGAT is the best existing graph-based model designed for dialogue-level RE. However, we can see that BERT still outperforms DHGAT

TABLE 4.2: Performance of all models on DialogRE. σ denotes the standard deviation computed from five runs of each model.

Model	Dev set		Test set	
	$F1$ (σ)	$F1c$ (σ)	$F1$ (σ)	$F1c$ (σ)
CNN [63]	46.1 (0.7)	43.7 (0.5)	48.0 (1.5)	45.0 (1.4)
LSTM [64]	46.7 (1.1)	44.2 (0.8)	47.4 (0.6)	44.9 (0.7)
BiLSTM [65]	48.1 (1.0)	44.3 (1.3)	48.6 (1.0)	45.0 (1.3)
AGGCN [25]	46.6	40.5	46.2	39.5
LSR [61]	44.5	-	44.4	-
DHGAT [60]	57.7	52.7	56.1	50.7
BERT [10]	60.6 (1.2)	55.4 (0.9)	58.5 (2.0)	53.2 (1.6)
BERTs [1]	63.0 (1.5)	57.3 (1.2)	61.2 (0.9)	55.4 (0.9)
GDPNet (our model)	67.1 (1.0)	61.5 (0.8)	64.9 (1.1)	60.1 (0.9)

FIGURE 4.1: GDPNet and BERTs on different dialogue lengths.

by 2.4 and 2.5 in $F1$ and $F1c$. Among BERT-based models, BERTs is superior to BERT because BERTs incorporate speaker-related information, and BERTs is also the best existing model of dialogue-level RE. Following the same input format and setting, GDPNet is built on top of BERTs. That is, BERTs acts as the BERT module for feature extraction in GDPNet (see Figure 3.2). Shown in Table 4.2, GDPNet outperforms BERTs by 3.7 points in $F1$ on test set. In addition, we can also observe that our GDPNet has a better performance when there is no entire dialogue given because GDPNet achieved better performance than BERTs by 4.7 points on $F1c$. Last, when we run the code 5 times by different random seeds, the standard deviation of performance is closed to the baseline models, which means our model has similar stability with SoTA (*i.e.* BERTs).

GDPNet is designed to find key information from long sequences for effective RE.

The DTWPool in GDPNet is used to refine the latent graph and identify the informative nodes to improve the performance. Thus, we expect that GDPNet is capable of tackling long sequences better. To verify that our model can indeed handle longer sequences better, we group the dialogues in DialogRE test set into five subsets by their length, *i.e.*, number of tokens. On these subsets, we evaluate GDPNet and BERTs after training. Figure 4.1 reports $F1$ scores of GDPNet. BERTs has better performance than our model if the sequence is short, but our GDPNet consistently outperforms BERTs when dialogue length is more than 100 tokens. In particular, GDPNet surpasses BERTs by a large margin on the dialogues with over 400 tokens. This comparison shows that GDPNet is effective in modeling long sequences through refining the latent graph built on top of token representations.

4.1.5 Case Study

TABLE 4.3: Text of the first dialogue in test set of DialogRE. The tokens in bold are selected by GDPNet.

S1:	Hey, you guys! Look what I found! Look at this!
	That's my Mom's writing! Look.
S2:	Me and Frank and Phoebe, Graduation 1965.
S1:	Y'know what that means?
S3:	That you're actually 50?
S1:	No-no, that's not , that's not me Phoebe, that's her pal Phoebe. According to her high school year- book, they were like B.F.F. Best Friends Forever.
<hr/>	
	Argument Pair Trigger Relation type
R1	(S1, Frank) high school yearbook per:alumni

To further understand how does our GDPNet work, we provide a case study in this section. We randomly select one model within the five models we obtained during training, and we evaluate this model on the test set of dialogue-level RE dataset (*i.e.* *DialogRE*). We simply select the first sample in the test set as an example. In Table 4.3, when we are aiming at predicting the Relation 1 (*i.e.* *R1*), Speaker 1 (*i.e.* *S1*) and Frank are given as the argument pair. The ground truth label of this target relation is "per:alumni" in our task. We use bold font to represent the tokens selected in the final task-specific graph in our GDPNet. We can observe that there are only a small number of tokens are preserved although there are multiple

TABLE 4.4: An ablation study on GDPNet model. For models without GGG, we replaced GGG by multi-head attention to initialize the edges of the multi-view graph. For the model without DTWloss, we remove the second term in Equation 3.25. All DTWPool layers are removed for the models without DTWPool.

Model	$F1(\sigma)$	$F1c(\sigma)$
GDPNet	64.9 (1.1)	60.1 (0.9)
with Homogeneous GGG	63.5 (0.7)	58.4 (0.6)
w/o GGG	62.1 (1.6)	58.1 (1.1)
w/o DTWloss	63.4 (1.4)	58.6 (1.3)
w/o DTWPool	48.9 (1.1)	22.4 (1.0)
w/o GGG & DTWPool	48.2 (1.4)	21.8 (1.0)

turns with three speakers. If we check the selected tokens in detail, we can find that some informative tokens are selected precisely. For instance, in this sample, the trigger words representing the most important words provided by the dataset are also preserved during the graph pooling. For R1, the trigger words are *high school yearbook*. We can see that some informative tokens in these trigger words are selected (*e.g. school*). At the same time, a few other helpful tokens are identified by our model, such as *Frank* and *her*. Some less related tokens, like “you’re actually 50?”, are ignored by our model. At the same time, not all informative tokens are selected by our model, such as *high* and *yearbook*, although these two tokens are trigger words annotated. We suggest that the token representations from BERT module cover context information to some extent, so the semantic information of these tokens can be incorporated in the contiguous token representations from BERT-based model.

4.1.6 Ablation Study

We conduct ablative experiments on DialogRE to evaluate the effectiveness of the two main components in our GDPNet, *i.e.* Gaussian Graph Generator and Dynamic Time Warping Pooling. The results are reported in Table 4.4.

To verify the effectiveness of multi-view graph, we first replace the multi-view graph with a simple homogeneous graph, which is the same as setting the number of views to one. The homogeneous graph is still generated by GGG, we simply use one Gaussian distribution to represent one node. And we use the original

graph convolution instead of the multi-view graph convolution to update the node representations of the graph. The performance degradation suggests that multi-view graph is beneficial as it models complex relationships among tokens. Next, we evaluate GGG by replacing GGG with multi-head attention. Specially, we use multi-head attention instead of GGG to initialize the multi-view view graph, and we used the same approaches to process the graph generated. The results show that without GGG, performance drops. GGG initializes the latent graph by measuring the difference between two Gaussian distributions generated from node representations, which decouples the dependency between token representations and graph edges. To evaluate the impact of DTWPool, we first drop DTWloss by removing the second term in Equation 3.25, so the loss function only incorporates the first term (*i.e.* a cross-entropy loss or a binary cross-entropy loss). We can see a slight performance drop in Table 4.4. Without DTWloss, DTWPool degenerates into the multi-view version of SAGPool [74]. This result indicates that DTWPool outperforms SAGPool when tackling the multi-view graph learned from a token sequence. Finally, when all the DTWPool layers are removed, the performance of GDPNet decreases dramatically, which shows that DTWPool is crucial for the GDPNet. After removing both GGG and DTWPool, the performance of GDPNet is even worse. To summarize, DTWPool is crucial for learning a task-specific graph from a large latent multi-view graph. The final graph learned effectively filters out less useful information from a long sequence for effective relation extraction.

4.2 Sentence-level Relation Extraction

4.2.1 Datasets

We first evaluate GDPNet on two sentence-level RE datasets, TACRED [2] and TACRED-Revisit [27]. TACRED is a widely used large-scale sentence-level relation extraction dataset. It contains more than 106K sentences drawn from the yearly TACKBP4 challenge, and 42 different relations (41 common relation types and a special “no relation” type). The subject mentions in TACRED are person and organization, while object mentions are in 16 fine-grained types, including date, location, etc. We then conduct the experiments on TACRED-Revisit [27], which is a modified version of TACRED dataset. Since there are over 50 percent of labels

in TACRED dataset are required to be relabeled, authors released it recently and corrected the wrong labels in the development and test sets of TACRED.

4.2.2 Baseline Models

To the best of our knowledge, SpanBERT [11] is the best performing sentence-level RE model without incorporating any external knowledge and parser. We consider SpanBERT as a strong baseline to benchmark our GDPNet. For a comprehensive comparison, we also include RNN-based models and graph-based models in sentence-level RE. For RNN-based models, we include typical RNN-based baseline model (*i.e.* LSTM [39]) and attention-based models (*e.g.* PA-LSTM [2] and SA-LSTM [22]) in our experiments. We also incorporate a few graph-based models like C-GCN [24], C-AGGCN [25] and LST-AGCN [45]. We report the performance of other BERT-based models (*e.g.* ERNIE [54] and KnowBERT [12]) trained by external resources, although these models are not our direct baselines.

4.2.3 Experimental Setup

Similar to the setting of dialogue-level RE, we use the same input format and hyperparameter settings as in SpanBERT [11] because it is our strongest baseline. Specially, subject entity X_s and object entity X_o are each replaced by a sequence of “[SUBJ-NER]” or “[OBJ-NER]” tokens to leverage the overfitting. This is different from the settings in DialogRE where X_s and X_o are appended to the input sequence X . Then [CLS] X [SEP] forms the input to the models.

We still adopt Adam [79] to optimize our model, but the learning rate is $2e - 5$ in the sentence-level RE. Since there are more samples in TACRED, we suggest using a smaller epoch and larger batch size for faster training. The epoch is 10, and batch size is 32 in this part. The hyperparameter settings of graph module are similar but different. The same part is that we use the same number of GCN layers and DTWPool layers with dialogue-level RE in our evaluation. However, we set the pooling ratio lower bound as 0.8 for TACRED because the sentences are relatively short. Due to the different lengths of input context, we use different weights of SoftDTW loss. The weight of SoftDTW loss is set as $2e - 4$ in this work.

TABLE 4.5: Hyperparameter settings on TACRED.

Parameter	Value
Epoch	10
Batch Size	32
Learning rate	2e-5
Dropout	0.5
Hidden units of graph	300
Number of views	3
Number of DTWPool layers	3
Pooling ratio lower bound	0.8
Weight of SoftDTW loss	2e-4

TABLE 4.6: Performance of all models on TACRED.

Model	<i>Precision</i>	<i>Recall</i>	<i>F1</i>
LSTM [39]	65.7	59.9	62.7
PA-LSTM [2]	65.7	64.5	65.1
SA-LSTM [22]	68.1	65.7	66.9
SA-LSTM+D [22]	69.0	66.2	67.6
GCN [24]	69.8	59.0	64.0
C-GCN [24]	69.9	63.3	66.4
AGGCN [25]	69.9	60.9	65.1
C-AGGCN [25]	73.1	60.9	68.2
LST-AGCN [45]	-	-	68.8
BERT [10]	69.1	63.9	66.4
BERT-LSTM [52]	73.3	63.1	67.8
R-BERT [53]	-	-	69.4
SpanBERT [11]	70.8	70.9	70.8
GDPNet (Our model)	72.0	69.0	70.5
ERNIE [54]	69.9	66.0	67.9
KnowBERT [12]	71.6	71.4	71.5

4.2.4 Results on TACRED

We first report the result on TACRED dataset, since it is a standard dataset with more baselines on sentence-level relation extraction. The results are summarized in Table 4.6. Similar observations hold, that BERT-based models (*i.e.*, SpanBERT and KnowBERT), significantly outperform non-BERT models (*i.e.*, LSTM,

TABLE 4.7: Performance of all models on TACRED-Revisit. we run the released code if available, and mark results obtained by asterisk(*). For the models without reported performance on TACRED-Revisit, we run the released code if available.

Model	<i>Precision</i>	<i>Recall</i>	<i>F1</i>
LSTM [39]	71.5*	69.7*	70.6*
PA-LSTM [2]	74.5*	74.1*	74.3*
C-AGGCN [25]	77.7*	73.4*	75.5*
SpanBERT [11]	75.7*	80.7*	78.0
GDPNet (Our model)	79.4	81.0	80.2
KnowBERT [12]	-	-	79.3

PA-LSTM, C-AGGCN and LST-AGCN) on TACRED. GDPNet achieves comparable performance with SpanBERT³ on TACRED. Compared to KnowBERT, which utilize external knowledge in its training, GDPNet’s *F1* is lower by 1 point on TACRED. Compared to dialogue, sentence is much shorter and BERT-based models are effective in capturing the key information. As GDPNet is designed for handling long sequences, we do not expect it to outperform SoTA models, but GDPNet remains competitive for sentence-level RE.

4.2.5 Results on TACRED-Revisit

Since the annotations of TACRED dataset are debatable, we also evaluate our GDPNet on TACRED-Revisit. The performance is reported in Table 4.7. Compared with the results on TACRED, our GDPNet achieves better results on TACRED-Revisit. We can observe our model outperforms SpanBERT by a large margin. In addition, although our model is trained without external resources, GDPNet outperforms KnowBERT by almost 1 point on TACRED-Revisit.

TABLE 4.8: Percentage (%) of the tokens selected in the final graph from sequence, *e.g.* an entire dialogue in DialogRE or the whole sentence in TACRED. Non-repetitive tokens are the tokens that appear only once in the sequence; repetitive tokens appear two or more times in the sequence. Trigger tokens are key tokens annotated in DialogRE.

Type of tokens	DialogRE	TACRED
All tokens	15.6	66.3
Non-repetitive tokens	23.5	67.6
Repetitive tokens	10.0	58.1
Trigger tokens	32.1	-

4.3 Quantitative Analysis

Our last experiment is to analyze DTWPool in GDPNet. Our DTWPool is designed to identify indicative tokens for relation extraction through refining the latent multi-view graph. To verify that our DTWPool can handle these sparse informative tokens well, we conduct a quantitative analysis in this section. Table 4.8 reports the percentage of tokens selected in the final graph after the DTWPool process, on both DialogRE and TACRED datasets. We separate the repetitive tokens and non-repetitive tokens based on the original input, *i.e.* whether the word appears only once or multiple times in the input sequence. Repetitive tokens, in general, define the topic of the sentence or dialogue. However, the relation type between two entities is seldom described repetitively. In fact, given the same dialogue, we may predict different relation types between different pairs of entities. With this in mind, we consider repetitive tokens are less important compared to non-repetitive tokens for RE tasks in general. As shown in Table 4.8, DTWPool selects more non-repetitive tokens than repetitive tokens on both datasets, in particular, on the DialogRE dataset. More interestingly, DialogRE provides manually annotated trigger tokens that are indicative of the relation type. DTWPool selects 32.1% of trigger tokens, given that only 15.6% of tokens are selected among all tokens. That is, trigger tokens are selected with a much higher chance than random. This analysis shows that DTWPool is capable of selecting indicative tokens for relation extraction.

³A very recent study [59] also combines SpanBERT with GCN, but it still relies on an external parser for graph generation. In contrast, our GDPNet regards the graph as a latent variable, which is more general and feasible.

4.4 Summary

We evaluated GDPNet on two relation extract tasks, dialogue-level RE and sentence-level RE in this chapter. On dialogue-level RE, our model achieved the best performance on DialogRE dataset. On sentence-level RE, our model also achieved comparable performance on TACRED and outperforms SoTAs without external training resources on TACRED-Revisit. Finally, we conduct a quantitative analysis, which is designed to justify that our model can identify the sparse indicative tokens in an unsupervised manner. The result matches well with our expectations. All these experimental results show the effectiveness of our model.

Chapter 5

Conclusion and Future Work

In this thesis, our goal is to improve relation extraction. To this end, we propose GDPNet to find indicative words from long sequences (*e.g.* dialogues) for effective relation extraction. In the first chapter, Chapter 1 - *Introduction*, we introduce the background of our research problem and its potential application on other downstream tasks. The definition of relation extraction, the research focus of this thesis and the meaning of research relation extraction are also given in Chapter 1. We highlight the motivation of our research problem and present our solution briefly.

Chapter 2, *Literature Review* provides the readers with background information about existing approaches on sentence-level RE and dialogue-level RE. We mainly review the problem definition, datasets and related existing approaches of the two tasks we are interested in. We also analyze the challenges in relation extraction and provide our solution in the following chapters.

Chapter 3, *Approach* presents our model, GDPNet, which is designed for identifying sparse indicative words to improve relation extraction by refining the latent multi-view graph. We first introduce the overall structure of our model. In general, GDPNet incorporates BERT module, graph module and a final classifier. Specifically, we first use the BERT module as feature extractor to encode context representations for each token in the input text. The BERT-based model can be different on different relation extraction tasks. In this thesis, we use SpanBERT as feature encoder for sentence-level RE, and BERTs is used for dialogue-level RE.

The token representations from BERT module except sequence representation corresponding to [CLS] token are employed as the input of graph module. The input token representations are used as nodes to construct a latent multi-view graph by GGG proposed. To further refine the latent multi-view graph, we propose DTW-Pool, a graph pooling method to refine the multi-view graph learned from sequence. The graph is updated through multiple interactions of multi-view graph convolution and DTWPool. Finally, the refined latent graph is fed into the classifier to predict relation type. We expect our model is able to improve relation extraction by identifying sparse informative tokens.

Chapter 4, *Experiments* aims to verify the effectiveness of GDPNet. We conduct two sets of experiments in this chapter to compare GDPNets to SoTAs on dialogue-level RE and sentence-level RE, respectively. We first compare with SoTAs on DialogRE, a dialogue-level RE dataset, and our GDPNet achieved the best performance on dialogue-level RE. We then provide a case study to show how does our model work. An ablation study is also performed on DialogRE to verify the effectiveness of the two main components in our GDPNet, *i.e.* Gaussian Graph Generator and Dynamic Time Warping Pooling. In addition, we evaluate our model on TACRED and TACRED-Revisit, a widely sentence-level RE dataset and its modification. GDPNet achieved comparable performance on TACRED and outperforms SoTAs without external training resources on TACRED-Revisit. Last, we conduct a quantitative study to analyze DTWPool in GDPNet. The result shows that DTWPool is capable of selecting indicative tokens for relation extraction in an unsupervised manner.

5.1 Future Work

The relations are treated independently in GDPNet, but the relations are related to each other in dialogue-level RE. We can extend GDPNet to handle interrelations among multiple relations. One possible solution is connecting the latent graphs of different relations, and then we can reason the interrelation among different relations in the whole graph. We plan to use the task-specific graph of each relation because the final graph would be much larger during the connection. Chapter 4 has shown that the task-specific graph in GDPNet can identify the sparse informative tokens to represent one relation by a graph with only a few nodes. We

can then concatenate different task-specific graphs according to the shared entity, which means the task-specific graph of relations sharing the same entity would be connected in the final graph. In this thesis, GCN has shown its great performance in reasoning, so we can still use GCN to reason the interrelation among relations. We believe this approach can produce a significant gain in the dialogue-level RE performance.

Another potential direction is improving our GDPNet by knowledge graph. Currently, we train our model without any external resources including knowledge base. However, on sentence-level RE, knowledge graph has shown it can improve the performance significantly. For our GDPNet, we learn a latent graph from text, and in the future, we can use knowledge graph to guide the Gaussian Graph Generator to generate the edges. We use Gaussian distributions to model each token representation in this thesis. Similarly, we can use Gaussian distribution to model each node in knowledge graph, and we then align our graph and the knowledge graph. One potential way is adopting KL divergence between the Gaussian distribution from latent graph and the Gaussian distribution from knowledge graph to measure the difference for alignment. In this way, the knowledge is taken into consideration to improve relation extraction model.

List of Author's Awards, Patents, and Publications¹

Conference Proceedings

- **Fuzhao Xue**, Aixin Sun, Hao Zhang, Eng-Siong Chng "GDPNet: Refining Refining Latent Multi-View Graph for Relation Extraction" Published at Association for the Advancement of Artificial Intelligence (AAAI) 2021

¹The superscript * indicates joint first authors

Bibliography

- [1] Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. Dialogue-based relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4927–4940, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.444. URL <https://www.aclweb.org/anthology/2020.acl-main.444>. xvii, 1, 3, 5, 6, 10, 18, 19, 21, 22, 23, 31, 44, 45, 46
- [2] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. xvii, 1, 2, 4, 6, 10, 13, 14, 15, 17, 49, 50, 51, 52
- [3] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007. 1
- [4] Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2118. URL <https://www.aclweb.org/anthology/N18-2118>. 1
- [5] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. A survey on knowledge graphs: Representation, acquisition and applications. *arXiv preprint arXiv:2002.00388*, 2020. 1
- [6] Debashis Das, Laxman Sahoo, and Sujoy Datta. A survey on recommendation system. *International Journal of Computer Applications*, 160(7), 2017. 1
- [7] Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, and Wei Wang. Kbqa: Learning question answering over qa corpora and knowledge bases. *Proc. VLDB Endow.*, 10(5):565–576, January 2017. ISSN 2150-8097. doi: 10.14778/3055540.3055549. URL <https://doi.org/10.14778/3055540.3055549>. 1

- [8] Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1074. URL <https://www.aclweb.org/anthology/P19-1074>. 1, 10
- [9] Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium, October–November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1514. URL <https://www.aclweb.org/anthology/D18-1514>. 1, 10
- [10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL <https://www.aclweb.org/anthology/N19-1423>. 2, 4, 17, 21, 22, 23, 31, 32, 46, 51
- [11] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020. 2, 5, 16, 17, 18, 31, 33, 50, 51, 52
- [12] Matthew E. Peters, Mark Neumann, Robert L Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. Knowledge enhanced contextual word representations. In *EMNLP*, 2019. 2, 16, 17, 50, 51, 52
- [13] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhiyuan Liu, Juanzi Li, and Jian Tang. Kepler: A unified model for knowledge embedding and pre-trained language representation. *arXiv preprint arXiv:1911.06136*, 2019. 2, 17
- [14] Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Cuihong Cao, Daxin Jiang, Ming Zhou, et al. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*, 2020. 2, 17, 18
- [15] Junwei Bao, Nan Duan, Ming Zhou, and Tiejun Zhao. Knowledge-based question answering as machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 967–976, 2014. 3

- [16] Martin Krallinger, Obdulia Rabal, Saber A Akhondi, Martín Pérez Pérez, Jesús Santamaría, GP Rodríguez, et al. Overview of the biocreative vi chemical-protein interaction track. In *Proceedings of the sixth BioCreative challenge evaluation workshop*, volume 1, pages 141–146, 2017. 3
- [17] Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*, 2020. 3
- [18] Lin Xu, Qixian Zhou, Ke Gong, Xiaodan Liang, Jianheng Tang, and Liang Lin. End-to-end knowledge-routed relational dialogue system for automatic diagnosis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7346–7353, 2019. 3
- [19] Bryan Rink and Sanda Harabagiu. UTD: Classifying semantic relations by combining lexical and semantic resources. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 256–259, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/S10-1057>. 4, 14
- [20] Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1785–1794, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1206. URL <https://www.aclweb.org/anthology/D15-1206>. 4
- [21] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-2034. URL <https://www.aclweb.org/anthology/P16-2034>. 4, 19, 20
- [22] Bowen Yu, Zhenyu Zhang, Tingwen Liu, Bin Wang, Sujian Li, and Quangan Li. Beyond word attention: Using segment attention in neural relation extraction. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5401–5407. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/750. URL <https://doi.org/10.24963/ijcai.2019/750>. 4, 14, 17, 50, 51
- [23] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing,

- China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1150. URL <https://www.aclweb.org/anthology/P15-1150>. 4
- [24] Yuhao Zhang, Peng Qi, and Christopher D. Manning. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium, October–November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1244. URL <https://www.aclweb.org/anthology/D18-1244>. 4, 15, 17, 50, 51
- [25] Zhijiang Guo, Yan Zhang, and Wei Lu. Attention guided graph convolutional networks for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 241–251, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1024. URL <https://www.aclweb.org/anthology/P19-1024>. 4, 15, 17, 20, 22, 26, 36, 44, 45, 46, 50, 51, 52
- [26] Marco Cuturi and Mathieu Blondel. Soft-DTW: a differentiable loss function for time-series. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 894–903, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. 6, 28
- [27] Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. Tacred revisited: A thorough evaluation of the tacred relation extraction task. In *Proceedings of ACL*, 2020. URL <https://arxiv.org/abs/2004.14855>. 6, 13, 49
- [28] Zhijiang Guo, Guoshun Nan, Wei Lu, and Shay B Cohen. Learning latent forests for medical relation extraction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 3651–3675, 2020. 10, 11
- [29] Chang Wang and James Fan. Medical relation extraction with manifold models. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 828–838, 2014. 10, 11
- [30] Xiaozhi Wang, Xu Han, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Adversarial multi-lingual neural relation extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1156–1166, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/C18-1099>. 10, 11
- [31] Dan Roth and Wen-tau Yih. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 1–8, Boston, Massachusetts, USA, May 6 - May 7 2004. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/W04-2401>. 11

- [32] George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA). URL <http://www.lrec-conf.org/proceedings/lrec2004/pdf/5.pdf>. 12
- [33] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/S10-1006>. 12
- [34] Qiongxing Tao, Xiangfeng Luo, Hao Wang, and Richard Xu. Enhancing relation extraction using syntactic indicators and sentential contexts. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (IC-TAI)*, pages 1574–1580. IEEE, 2019. 12
- [35] Cheng Li and Ye Tian. Downstream model design of pre-trained language model for relation extraction task. *arXiv preprint arXiv:2004.03786*, 2020. 12
- [36] Hengguan Huang, Fuzhao Xue, Hao Wang, and Ye Wang. Deep graph random process for relational-thinking-based speech recognition. In *International Conference on Machine Learning*, pages 4531–4541. PMLR, 2020. 14
- [37] Huy Phan, Philipp Koch, Fabrice Katzberg, Marco Maass, Radoslaw Mazur, and Alfred Mertins. Audio scene classification with deep recurrent neural networks. *arXiv preprint arXiv:1703.04770*, 2017. 14
- [38] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL <https://www.aclweb.org/anthology/D14-1179>. 14
- [39] Patrick Verga, David Belanger, Emma Strubell, Benjamin Roth, and Andrew McCallum. Multilingual relation extraction using compositional universal schema. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 886–896, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1103. URL <https://www.aclweb.org/anthology/N16-1103>. 14, 17, 19, 50, 51, 52

- [40] Jingyu Zhao, Feiqing Huang, Jia Lv, Yanjie Duan, Zhen Qin, Guodong Li, and Guangjian Tian. Do rnn and lstm have long memory? *arXiv preprint arXiv:2006.03860*, 2020. 14
- [41] Hai Ye, Zichao Yan, Zhunchen Luo, and Wenhan Chao. Dependency-tree based convolutional neural networks for aspect term extraction. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 350–362. Springer, 2017. 15
- [42] Timothy Dozat and Christopher D Manning. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*, 2016. 15
- [43] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr, Christopher Fifty, Tao Yu, and Kilian Q Weinberger. Simplifying graph convolutional networks. *arXiv preprint arXiv:1902.07153*, 2019. 15
- [44] Zhijiang Guo, Yan Zhang, Zhiyang Teng, and Wei Lu. Densely connected graph convolutional networks for graph-to-sequence learning. *Transactions of the Association for Computational Linguistics*, 7:297–312, March 2019. doi: 10.1162/tacl_a_00269. URL <https://www.aclweb.org/anthology/Q19-1019>. 15, 36, 41
- [45] Kai Sun, Richong Zhang, Yongyi Mao, Samuel Mensah, and Xudong Liu. Relation extraction with convolutional network over learnable syntax-transport graph. In *AAAI*, pages 8928–8935, 2020. 15, 17, 50, 51
- [46] Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. A walk-based model on entity graphs for relation extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 81–88, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2014. URL <https://www.aclweb.org/anthology/P18-2014>. 16
- [47] Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4925–4936, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1498. URL <https://www.aclweb.org/anthology/D19-1498>.
- [48] Kazuma Hashimoto and Yoshimasa Tsuruoka. Neural machine translation with source-side latent graph parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 125–135, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1012. URL <https://www.aclweb.org/anthology/D17-1012>. 16

- [49] Mihai Masala, Stefan Ruseti, and Mihai Dascalu. Robert—a romanian bert model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6626–6637, 2020. [16](#)
- [50] Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A Gers. How does bert answer questions? a layer-wise analysis of transformer representations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1823–1832, 2019.
- [51] Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. Multi-passage bert: A globally normalized bert model for open-domain question answering. *arXiv preprint arXiv:1908.08167*, 2019. [16](#)
- [52] Peng Shi and Jimmy Lin. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*, 2019. [16](#), [17](#), [51](#)
- [53] Shanchan Wu and Yifan He. Enriching pre-trained language model with entity information for relation classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2361–2364, 2019. [16](#), [17](#), [51](#)
- [54] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In *Proceedings of ACL 2019*, 2019. [16](#), [17](#), [50](#), [51](#)
- [55] Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1279. URL <https://www.aclweb.org/anthology/P19-1279>. [16](#), [17](#)
- [56] Yi-Ting Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. MPDD: A multi-party dialogue dataset for analysis of emotions and interpersonal relationships. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 610–614, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL <https://www.aclweb.org/anthology/2020.lrec-1.76>. [18](#), [19](#)
- [57] Qi Jia, Hongru Huang, and Kenny Q Zhu. Ddrel: A new dataset for interpersonal relation classification in dyadic dialogues. *arXiv preprint arXiv:2012.02553*, 2020. [18](#), [19](#)
- [58] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL <https://www.aclweb.org/anthology/D14-1162>. [19](#)

- [59] Jun Chen, Robert Hoehndorf, Mohamed Elhoseiny, and Xiangliang Zhang. Efficient long-distance relation extraction with dg-spanbert. *arXiv preprint arXiv:2004.03636*, 2020. [20](#), [53](#)
- [60] Hui Chen, Pengfei Hong, Wei Han, Navonil Majumder, and Soujanya Poria. Dialogue relation extraction with document-level heterogeneous graph attention networks. *arXiv preprint arXiv:2009.05092*, 2020. [20](#), [21](#), [22](#), [23](#), [44](#), [46](#)
- [61] Guoshun Nan, Zhiqiang Guo, Ivan Sekulic, and Wei Lu. Reasoning with latent structure refinement for document-level relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1546–1557, Online, July 2020. ACL. [20](#), [22](#), [44](#), [46](#)
- [62] Terry Koo, Amir Globerson, Xavier Carreras, and Michael Collins. Structured prediction models via the matrix-tree theorem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 141–150, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/D07-1015>. [21](#)
- [63] S. Lawrence, C. L. Giles, Ah Chung Tsoi, and A. D. Back. Face recognition: a convolutional neural-network approach. *IEEE Transactions on Neural Networks*, 8(1):98–113, 1997. [22](#), [46](#)
- [64] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. [22](#), [44](#), [46](#)
- [65] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5–6):602–610, June 2005. [22](#), [44](#), [46](#)
- [66] H. Sakoe and Seibi Chiba. A dynamic programming approach to continuous speech recognition. 1971. [28](#)
- [67] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978. doi: 10.1109/TASSP.1978.1163055. [28](#)
- [68] Marco Cuturi, Jean-Philippe Vert, Oystein Birkenes, and Tomoko Matsui. A kernel for time series based on global alignments. *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, Apr 2007. doi: 10.1109/icassp.2007.366260. URL <http://dx.doi.org/10.1109/ICASSP.2007.366260>. [28](#)
- [69] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June

2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL <https://www.aclweb.org/anthology/N18-1202>. 31
- [70] A. Radford. Improving language understanding by generative pre-training. 2018. 31
- [71] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 34
- [72] Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1159. URL <https://www.aclweb.org/anthology/D17-1159>. 36
- [73] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. 37
- [74] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. Self-attention graph pooling. *arXiv preprint arXiv:1904.08082*, 2019. 38, 49
- [75] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc., 2017. URL <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>. 38
- [76] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 38
- [77] Hongyang Gao and Shuiwang Ji. Graph u-nets. *arXiv preprint arXiv:1905.05178*, 2019. 38
- [78] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in Neural Information Processing Systems*, volume 31, pages 4800–4810. Curran Associates, Inc., 2018. 38
- [79] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 45, 50