Multi-feature Fusion for Relation Extraction using Entity Types and Word Dependencies

Pu Zhang[1], Junwei Li[2], Sixing Chen[3], Jingyu Zhang[4], Libo Tang[5]
School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China[1, 2, 4]
Faculty of Engineering, School of Computer Science, University of Sydney, Sydney, Australia[3]
School of Software, Chongqing Institute of Engineering, Chongqing, China[5]

***Abstract*—Most existing methods do not make full use of different types of information sources to extract effective features for relation extraction. This paper proposes a multi-feature fusion model based on raw input sentences and external knowledge sources, which deeply integrates diverse lexical, semantic, and syntactic features into deep neural network models. Specifically, our model extracts lexical features of different granularity from the original input text representation, entity type features from the entity annotation information of the corpus, and dependency features from the dependency trees. Meanwhile, the dimension-based attention mechanism is proposed to enrich the diversity of entity type features and enhance their discriminability. Different features enable the model to comprehensively utilize various types of information, so this paper fuses these features and train a classifier for relation extraction. The experimental results show that the proposed model outperforms the existing state-of-the-art baselines on the TACRED Revisited, Re-TACRED, and SemEval datasets, with macro-average F1 scores of 81.2%, 90.2%, and 89.4%, respectively, improving the performance by 1.4%, 4.4%, and 2% on average, which indicates the effectiveness of multi-feature fusion modeling.**

***Keywords*—Relation extraction; multi-feature fusion; information extraction; dependency tree; entity type**

I. Introduction

Relation extraction (RE) aims to extract the relationships between entities from free text [1], which can provide support for high-level tasks including knowledge graph construction [2], text summarization [3], question answering [4], and so on. As an important and challenging task, RE has recently received considerable attention from researchers. Specifically, neural relation extraction (NRE) models have emerged and achieved promising performance thanks to the remarkable advancement of deep learning [5-7].

It is essential to fully exploit the different types of features to enhance the performance of the RE task. To utilize rich lexical information in the word sequences, many NRE models have been proposed to extract lexical features, including convolutional neural network (CNN) based [8], recurrent neural network (RNN) based [9], recursive neural network (Recursive NN) based [10], and transformer based [11] models. Most recently, without using any external tools or knowledge, Liang et al. [12] proposed a new model that extracts features from the original input sentences at entity mention, segment, and sentence levels. These methods focus on utilizing the overall or local information within word sequences but do not leverage more specific external knowledge, which may hinder performance of the model.

Besides learning lexical features from the raw data, many recent works also use external knowledge including knowledge graphs [13-14], dependency trees [15], and entity types [16] to construct explicit structured features. To infuse prior knowledge from the existing knowledge graphs, some works [13-14] have tried to integrate large-scale pre-trained models with knowledge bases (KBs) and use the models on numerous downstream tasks. Chen et al. [15] proposed a method that encodes and weights the dependency information by utilizing type-aware map memories (TaMM), which achieved outstanding results on the SemEval dataset [17]. Vashishth et al. [16] improved the performance of the RE model by enriching the features with additional entity type information in the graph structure. However, these methods only use a single form of external knowledge and do not encompass the collaborative use of multiple forms of external knowledge.

Despite their effectiveness, existing methods have the following drawbacks:

1) There are various types of information that can contribute to RE tasks, for example, the word sequences can be a source of rich lexical information, the dependency trees can provide syntactic information, and the entity types can provide constraint information of semantic relations over the entities. However, most of the previous works did not take them into account simultaneously and cannot take full advantage of different types of information sources to extract effective features.

2) A specific relation often constrains the entity types of its target entities. For instance, the place-of-birth relation restricts the entity types of a pair of entities to person and location, respectively. Therefore, entity types are important indicators for a specific relation. However, NRE models usually ignore such auxiliary information without using entity type information to impose constraints when extracting relations. Although a few studies have integrated entity type information into relation extraction, they are resource-centric and highly dependent on knowledge bases [16, 18]. Moreover, previous works often combined the coarse entity types of entity mentions with its contextual features, which suffers from coarse-grained entity types as they may fail to distinguish the relations.

To tackle these limitations, a multi-feature fusion model is proposed for RE. The model exploits both raw text data and external knowledge sources to obtain different types of features, filling the gap left by previous methods that did not simultaneously leverage both word sequences and multiple types of external knowledge. In detail, the model constructs representative original input features from the raw data, and obtains entity type and dependency features from external knowledge sources including entity annotation information in the corpus and dependency tree, respectively. Furthermore, the model employs a dimension-based attention mechanism to improve the diversity and discriminability of entity type features extracted from coarse-grained entity type information, addressing the issue of previous models being unable to distinguish between different relations. Finally, considering that different granularity features have complementary effects, we further fuse these features into a single vector via concatenation and perform relation extraction. The experimental results on the three public datasets demonstrate the effectiveness of our model.

Our contributions are summarized as follows:

3) We propose a multi-feature fusion model for relation extraction. To strengthen the ability to capture different kinds of features with various granularities, the model deeply integrates representative original input features with extra knowledge such as entity type information and dependency information, which can significantly boost the model's performance.

4) We present a dimension-based attention mechanism to enrich the diversity of the entity type features and enhance their discriminability, thus solving the problem of the coarse entity types of entity mentions.

5) We also carry out extensive experiments on the three public datasets. The results verify the benefits of multi-feature fusion modeling, and our model achieves significant improvements over competitive baselines.

The rest of this paper is structured as follows: Section II provides a review of related works; Section III presents the task definition, and Section IV provides the research objective; Section V describes in detail the proposed model; Section VI and VII discuss the experimental setups and results, and Section VIII concludes the paper.

II. Related Works

Early works on RE were mainly based on statistical machine learning. Kambhatla [19] combined a variety of features with a maximum entropy model for relation classification. Zhou et al. [20] incorporated semantic information into the feature-based relation extraction model to further boost the performance. Overall, these works require a significant level of manual design, and the quality of the hand-crafted features has a significant impact on the model's effectiveness.

With the maturity of deep learning technology, neural networks can automatically learn the potential features in a sentence and have been widely adopted in relation extraction tasks. Existing NRE models can be broadly classified into two categories: sequence-based and dependency-based [21].

Sequence-based models work with word sequences and concentrate on encoding the context information of a sentence by neural networks to capture latent features. Many models using various neural network architectures have been proposed to extract effective lexical features from the input. As CNN has achieved competitive performance on many traditional NLP tasks, Zeng et al. [8] employed it to extract features that contain valid lexical information for RE. Nguyen and Grishman [22] designed CNN models with convolutional kernels of multiple window sizes that can automatically learn implicit features in sentences, minimizing the reliance on external toolkits and resources. Zhang and Wang [10] employed RNN to model sentence context, allowing the model to capture both long-term and temporal features for RE. In order to extract multi-type features from input sentences, Wen et al. [23] combined the gate mechanism with the piecewise CNN to capture the features of the sentence.

Dependency-based models, as opposed to sequence-based models, use dependency parsing information to extract syntactic relations. Using dependency trees in RE has become a mainstream trend [24-25]. However, most dependency trees are generated by tools, which will cause a certain amount of noise, so efficient pruning methods are necessary. There are many pruning methods, which can rely on graph neural networks for key information selection [25-26], or specific attention mechanisms to dynamically select the important dependency information [15].

With the recent advancements in pre-trained language models (PLMs), the latest studies often employed popular models such as BERT [27] or XLNet [28] for RE tasks. Hou et al. [29] directly applied BERT to relation extraction and proposed a BERT-based model. Based on the bidirectional transformer, Yamada et al. [30] built a model to obtain contextualized representations of words and entities by treating them as independent tokens. Joshi et al. [31] extended BERT and proposed the SpanBERT model for span selection tasks. Wang et al. [32] used external knowledge to fine-tune the pre-trained model. Overall, the above-mentioned PLMs-based models have achieved promising success for the RE task.

Although the above studies have made significant progress in the field of relational extraction, however, they still have some shortcomings. Some of them [12,22] only use the original input to extract lexical features and fail to make comprehensive use of different information sources, while some of them [13-14] use the knowledge base to extract entity type features, which requires a large-scale external resource for support. Other studies [24-26] use dependency trees to obtain dependency features, but the pruning methods are quite complex.

Different from the existing NRE models, our model provides several feature extractors to explore various information sources and deeply integrates diverse lexical, syntactic, and semantic features in RE tasks. The model comprehensively uses the original input text, entity annotation information, and dependency information to extract features, and the extraction of entity type features is done in a way without relying on external knowledge bases. In addition, the extraction method of dependency features is simple and effective. In summary, the model is a multi-feature fusion model, which can not only effectively utilize various types of features to improve the model's performance but also has the advantage of the low computational cost of feature extraction. To the best of our knowledge, few previous studies have attempted this.

III. Task Definition

We define the sentence-level relation extraction task discussed in this work as follows. Let $x=\{x_1,x_2,\ldots,x_n\}$ be tokens of input. Let e_1 and e_2 be a pair of entities in the sentence. The RE task will learn a function $\mathrm{P}(r)= \mathrm{f}_\theta(x,e_1,e_2)$, where $r \in R$ and R is a pre-defined relation set.

IV. Research Objective

The objective of this paper is to address the following problems with RE: 1. It is necessary to make full use of the original input text and external information sources to effectively build the relationship extraction model. 2. The cost of introducing information sources should not be too high, and the model should not become heavily resource-dependent. 3. The method of extracting features from external information sources should be simple and effective. To achieve the above goals, we propose a multi-feature fusion model for relation extraction, which explores various information sources and investigates the incorporation of diverse lexical, syntactic, and semantic features in relation extraction. In our model, different from the existing works, the external information sources are easily accessible and do not depend on the large knowledge base, which can reduce time and space requirements and can be flexibly applied to more scenarios.

V. Proposed Model

The motivation of our model is to take full advantage of different types of information sources and fully exploit various types of features to improve performance. Fig. 1 depicts the model's structure as well as details of each component. The model is made up of three components: 1) origin input feature extractor; 2) entity type feature extractor; 3) dependency feature extractor.

The origin input feature extractor is responsible for capturing multi-granularity hierarchical features from the raw input sentences, including sentence level, segment level, and entity mention level features. Specifically, as the multi-granularity feature extractor named SMS proposed in [12] has achieved remarkable performance, we directly use it to construct the original input features and concatenate them with entity type information and dependency information. The entity type feature extractor follows the design of the key-value memory network (KVMN) [33] by constructing two memory slots to store the type information of the corresponding entities and then inputting the feature information of each slot into the model in combination with the dimension-based attention mechanism. The dependency feature extractor encodes dependency information obtained from the dependency parser. Finally, we classify the relation with a fully-connected layer by aggregating all the extracted features.

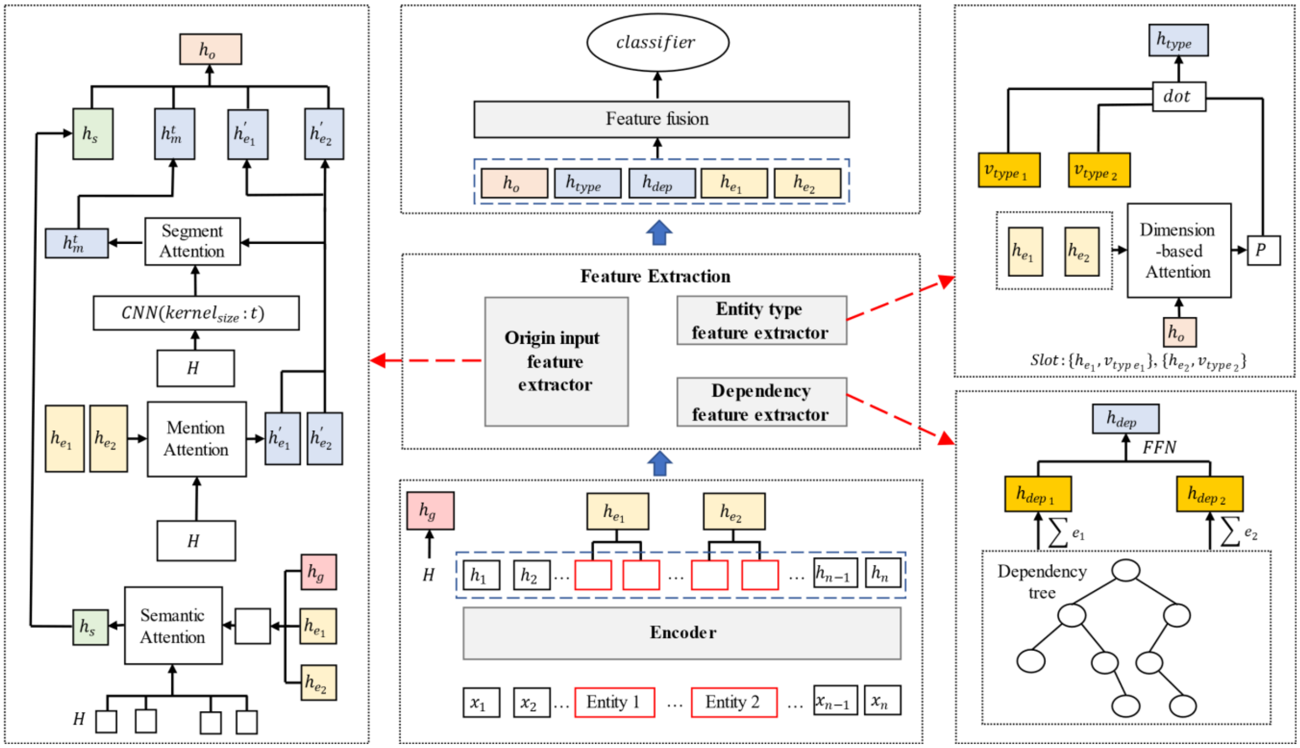


Fig. 1. The structure of our model.

In our model, three kinds of information sources are used to extract features, which are the original input sentences, the entity type's annotation information in the corpus, and the dependency trees of the sentences, respectively. Specifically, the original input sentences provide abundant lexical information, the dependency trees of the sentences carry long-distance syntactic information, and the entity type's information provides constraints information of the relation. For instance, in the sentence "A former Pakistani lawmaker has been arrested" with the marked entities "Pakistani" and "lawmaker", the relation between the two entities is "per:origin". To extract relation, the RE model needs to first capture lexical features of the sentence and the given entities, then catch entity type features and dependency features that are related to a specific relation, by combining the lexical with its syntactic and entity type features, the model can effectively model the contextual information required by RE task and predict the relation. In more detail, the entity types of two entities are helpful to capture the constraints of a specific relation and are important indicators for the relation. As shown in Fig. 2, if the types of the two entities are nationality and person, then there is more likely a "per:origin" or "org:founded_by" relation between the two entities than a "per:age" or "per:parents" relation. Moreover, considering that the dependency between the two entities is compound (compound expression), combined with the word sequence information and entity type information, the model will prefer to classify the relation as "per:origin" rather than "org:founded_by".

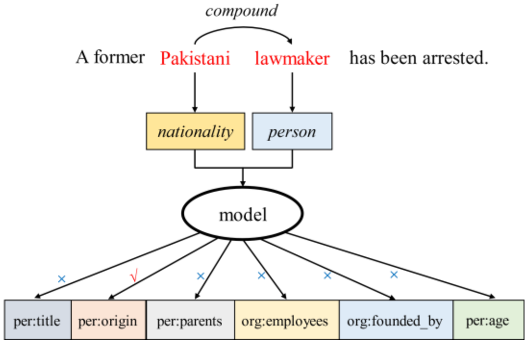


Fig. 2. Illustration of the relation extraction procedure for an example sentence.

A. Origin Input Feature Extractor

For the origin input feature extractor, we employ the SMS feature extractor proposed by Liang et al. [12], which fully exploits the input sentences to attain multi-granularity hierarchical features.

Firstly, a sequence of input tokens is transformed into vector representations using a BERT-related text encoder, which can be described as (1):

$$H = \{h_1, \dots, h_n\} = encoder(x_1, \dots, x_n)\# \tag{1}$$

Based on H, max-pooling operations can be used to get entity and sentence features, as shown in (2)-(3):

$$h_{e_1} = Maxpooling(h_{i:j}), h_{e_2} = Maxpooling(h_{k:l})\# \tag{2}$$

$$h_g = Maxpooling(H)\# \tag{3}$$

Where h_{e_1} and h_{e_2} are the representations of entity pairs, $(i:j)$ and $(k:l)$ are entity indices which delimit entity e_1 and e_2, and h_g is the input representation which captures global semantic information.

To obtain more information about entities e_1 and e_2 from input sentences, SMS utilizes a mention attention mechanism (Mention Attention in Fig. 1) to extract entity mention level features, as shown in (4):

$$h'_{e_i} = Softmax\left(\frac{H \cdot h_{e_i}}{\sqrt{d}}\right) \cdot H, i \epsilon \{1,2\}\# \tag{4}$$

Where h'_{e_1} and h'_{e_2} are entity mention features which capture more comprehensive entity information than h_{e_1} and h_{e_2} , and d denotes the dimension of vector representation.

To effectively capture the valuable local segments information, based on the n-gram segment level features $\{H_t\}_{t=1,2,3}$ extracted by CNN with different kernel sizes, SMS then utilizes a segment attention mechanism (Segment Attention in Fig. 1) to obtain mention-aware segment level features by combining the entity mention features h'_{e_1} and h'_{e_2} with H_t, which can be described as (5)-(6):

$$h_m^t = Softmax\left(\frac{H_t \cdot (W_m[h'_{e_1}; h'_{e_2}])}{\sqrt{d}}\right) \cdot H_t, t \epsilon \{1,2,3\}\# \tag{5}$$

$$H_t = CNN_t(H), t \epsilon \{1,2,3\}\# \tag{6}$$

Where t is CNN kernel size and H_t contains segment level features of 1,2,3-gram, and $\{h_m^t\}_{t=1,2,3}$ contain segments features with different granularity.

Then, SMS utilizes a global semantic attention operation (Semantic Attention in Fig. 1) which uses the concatenation of $[h_{e_1}; h_{e_2}; h_g]$ as the query to obtain the representation h_s, it contains sentence-level features related to entity mentions and captures deeper semantic features from the contextual representation H, as shown in (7).

$$h_s = Softmax\left(\frac{H \cdot (W_s[h_{e_1}; h_{e_2}; h_g])}{\sqrt{d}}\right) \cdot H\# \tag{7}$$

Where $W_s \epsilon \mathbb{R}^{d \times 3d}$ is a parameter matrix. Finally, SMS aggregates different-granularity features by (8).

$$h_o = ReLU(W_o[h_s; h'_{e_1}; h'_{e_2}; h_m^1; h_m^2; h_m^3])\# \tag{8}$$

Where $W_o \epsilon \mathbb{R}^{6d \times d}$ is a parameter matrix.

B. Entity Type Feature Extractor

The structure of the entity type feature extractor is shown in the upper right corner of Fig. 1. As a new neural network architecture, the key-value memory network (KVMN) [33] can effectively model pair-wisely organized information and has wide application scenarios in NLP tasks ([34-35]). Inspired by the architecture of KVMN, we also utilize a key-value structured memory and construct two memory slots to store the corresponding entity type information. Specifically, KVMN defines the memory slot s_i (i is the index of memory slots) as a pair of vectors $\{k_i, v_i\}$ where k_i is the key and v_i is the value, and stores the context information as a series of memory slots $s_i = \{k_i, v_i\}$. In our work, we build only two slots. $s_1 = \{e_1, entity\_type_1\}$ and $s_2 = \{e_2, entity\_type_2\}$ with *e* referring to the entity and *entity_type* referring to the entity type information. For the entity types which are used to compose features for training the model, there are only a few different types of entities, with entity types such as person or organization appearing more frequently than the remaining entity types such as date, money, etc., which leads to the model use coarse-grained entity types and often concentrates on a few common types, thus losing a certain amount of information diversity. In order to enrich the diversity of entity type features and increase the discriminability of entity type features, we design a dimension-based attention mechanism; it computes the entity type information in each memory slot to ensure that even if the entity types in different inputs are the same, they can have different effects on the model. Dimension-based means it calculates the attention scores for each dimension of the values in each memory slot.

For each memory slot, the keys and values are stored as $\{h_{e_1}, v_{type_1}\}$ and $\{h_{e_2}, v_{type_2}\}$, respectively, where h_{e_1} and h_{e_2} are the representations of entity pairs, v_{type_1} and v_{type_2} are the parameter vectors of the two entity types. The final representations are calculated as shown in (9)-(11).

$$p_i = softmax(W_e \cdot h_{e_i} \cdot h_o), i\epsilon\{1,2\}\# \tag{9}$$

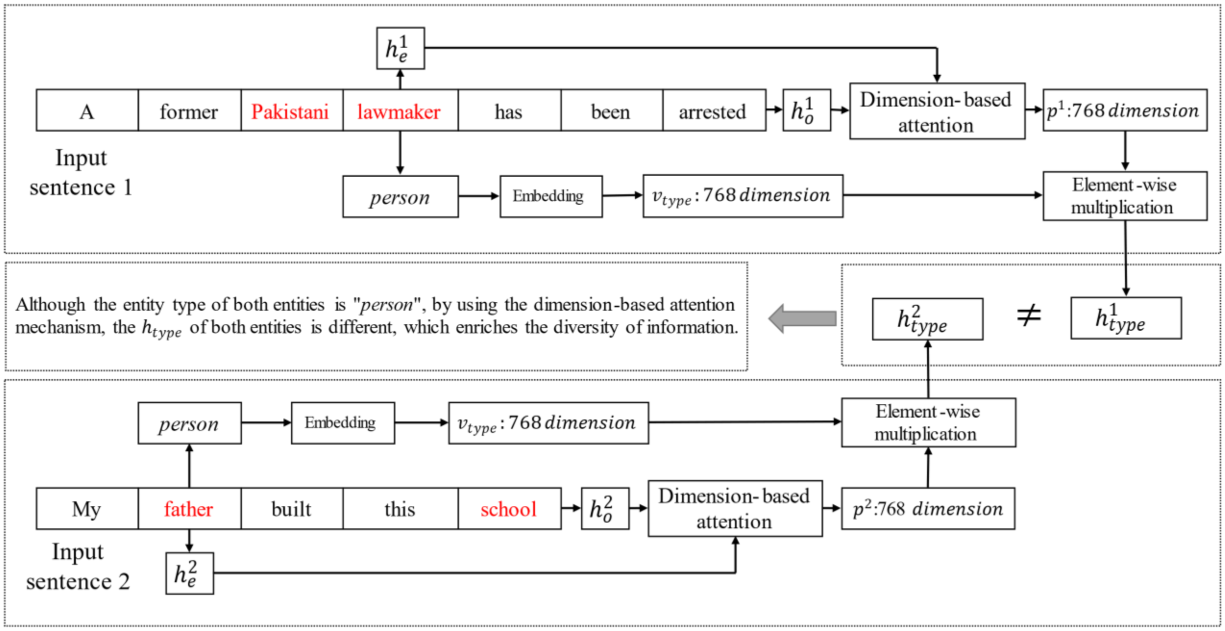
$$h_{type_i} = p_i \odot v_{type_i}, i\epsilon\{1,2\}\# \tag{10}$$

$$h_{type} = W_t\left[h_{type_1}; h_{type_2}\right]\# \tag{11}$$

where W_e, W_t are the corresponding parameter matrices, h_{e_i} are the shallow entity features introduced in Section V.A, h_o is the original input feature introduced in Section V.A, p_i indicates the importance score of each dimension of v_{ty} in a single memory slot and ⊙ denotes element-wise multiplication operation.

When utilizing entity type information, the original input features are used to obtain the importance weights of each dimension of the vector for the relevant entity type by using the dimension-based attention mechanism. An illustration of the mechanism is shown in Fig. 3.

In Fig. 3, for example, one sentence contains an entity “lawmaker”, and the other sentence has an entity “father”, both entities belonging to the same entity type “person”, and their parameter vectors of the entity type v_{type} are the same. Assuming that the dimension of the parameter vector is 768, then by combining h_e and h_o, a 768-dimensional weight vector p can be calculated, indicating the importance of each dimension of the parameter vector of the entity type. For entities, “lawmaker” and “father”, the shallow entity features h_e^1 and h_e^2 are different. For sentences 1 and 2, their original input features h_o^1 and h_o^2 are also different. As a result of the varied input contexts, the entity type features h_{type}^1 and h_{type}^2 are characterized differently, resolving the coarse-grained problem of entity types, increasing the diversity of entity type features, and improving the discriminability of entity type features.



Although the entity type of both entities is "*person*", by using the dimension-based attention mechanism, the h_{type} of both entities is different, which enriches the diversity of information.

Fig. 3. Illustration of the dimension-based attention mechanism.

C. Dependency Feature Extractor

The structure of the dependency feature extractor is shown in the lower right corner of Fig. 1. As the existing tools cannot ensure that the auto-generated dependency trees are totally right, a pruning strategy must be used to eliminate as much noise as possible while exploiting valid dependency information. We have tried several alternative approaches which use: 1) dependency information of the whole sentence; 2) dependency information of the whole sentence combined with the dimension-based attention mechanism; 3) dependency information directly related to the entities; 4) dependency information directly related to the entities with the dimension-based attention mechanism. Among these four approaches, approach three achieved good results, while the rest of the approaches failed to meet expectations. Therefore, when extracting the dependency feature, we simply use the dependency information directly related to the entities to prevent the introduction of too much noisy information. Referring to the KVMN, we also construct memory slots to record dependency information.

After getting the dependency parsing results of the input by using the toolkit such as Stanford CoreNLP Toolkit (SCT) or spaCy, for each word in the entity, its memory slot can be expressed as $s_i = \{k_i, v_i\}$, where k_i denotes the original word and v_i denotes the dependency type with its governor obtained from the dependency parse tree. In Fig. 4, as an example, the two entities are "*marrow bone*" and "*cell stem*" respectively, and their direct dependencies memory slot list should be S = [{*marrow*, *nsubj*},{*bone*, *compound*},{*cells*, *dobj*},{*stem*, *compound*}].

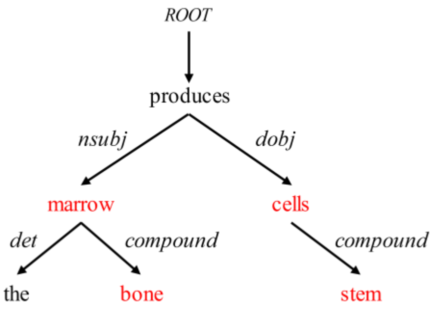


Fig. 4. Dependency tree of the sentence "the bone marrow produces stem cells".

More specifically, for each memory slot, the key is stored as the entity word vector obtained from the transformer encoder, the value is stored as a parameter vector which is obtained by using an embeddings lookup table, and the dependency features are calculated as shown in (12)-(13).

$$h_{dep_i} = \sum_{k=1}^{L_i} v_{dep_k^i}\, , i \epsilon \{1,2\}\# \tag{12}$$

$$h_{dep} = W_d\left[h_{dep_1}; h_{dep_2}\right]\# \tag{13}$$

Where L_i is the word count of entity i, W_d is the parameter matrix, $v_{dep_k^i}$ is the value vector obtained from the memory slot of the k-th word of entity i, h_{dep_1} and h_{dep_2} are the dependency features for the two entities. By combing the concatenation of h_{dep_1} and h_{dep_2} with a linear transform matrix, we can get the dependency feature denoted as h_{dep}.

D. Classification

Finally, we aggregate different types of features and output the relation label. In our model, we provide two feature fusion methods. The one method simply concatenates features, as shown in (14).

$$h = \left[h_o; h_{type}; h_{dep}; h_{e_1}; h_{e_2}\right]\# \tag{14}$$

The other method involves introducing a gating mechanism for the further fusion of various types of features, as shown in (15)-(17), where W_{type} and W_{dep} are parameter matrixes, b_{type} and b_{dep} are bias parameters, σ is a nonlinear activation function, and $\odot$ denotes element-wise multiplication.

$$h'_{type} = Gate\left(h_o, h_{type}\right) = \sigma\left(W_{type} h_{type} + b_{type}\right) \odot h_o\# \tag{15}$$

$$h'_{dep} = Gate\left(h_o, h_{dep}\right) = \sigma\left(W_{dep} h_{dep} + b_{dep}\right) \odot h_o\# \tag{16}$$

$$h = \left[h_o; h'_{type}; h'_{dep}; h_{e_1}; h_{e_2}\right]\# \tag{17}$$

Finally, we use softmax function to get the relation label, as shown in (18), where W is a trainable weight matrix and $|R|$ represents the number of relation labels.

$$\hat{r} = argmax \frac{exp(Wh_i)}{\sum_{i=1}^{|R|} exp(Wh_i)}\# \tag{18}$$

VI. Experiments

A. Datasets

We conduct experiments on SemEval 2010 Task 8 (SemEval) [17], TACRED Revisited (Tac-Rev) [36] and Re-TACRED [37] datasets to evaluate our model. Table I summarizes the statistics of the three datasets.

TABLE I. THE STATISTICS OF THE THREE DATASETS

Dataset	Training set	Validation set	Testing set	Relation types
SemEval[1]	8000	-	2717	19
Tac-Rev[2]	68124	22631	15509	42
Re-TACRED[3]	58465	19584	13418	40

Task 8 of SemEval-2010 aims to develop a standard testbed for future research and to provide a public dataset. The dataset contains 10,717 instances: 8,000 of them are released for training and the remainder is kept for testing. There are nine different types of relations in it, plus an additional "Other" type. When the two entities for each of the nine types of annotated relation types appear in the opposite order, it is implied that the phrase conveys the corresponding inverse relation for that type of relation. For example, the relations Entity-Destination(e1,e2) and Entity-Destination(e2,e1) are different from one another. Consequently, there are 19 different relation types in the SemEval dataset.

The TACRED Revisited (Tac-Rev) dataset is based on the original TACRED dataset [38]. Alt et al. [36] conducted an explorative analysis of the label quality for the TACRED dataset and found that a large fraction of the instances was incorrectly labeled by the crowd workers, and they corrected the errors in the Dev and Test sets.

Considering that the Tac-Rev dataset restricts revisions to a small subset of labels, and the majority of TACRED remains uncorrected. Stoica et al. [37] applied a better crowdsourcing strategy to re-annotate the entire TACRED dataset and then released Re-TACRED.

B. Settings

Detailed hyper-parameter settings for each dataset are shown in Table II.

TABLE II. THE HYPER-PARAMETER SETTINGS

	Tac-Rev	Re-TACRED	SemEval
Learning rate	3e-5	3e-5	3e-5
Warmup steps	300	300	-
Warmup rate	-	-	0.06
Epoch	4	4	10
Batch size	64	64	32

PyTorch is used to implement the proposed model. Following the official script, we use the Macro-F1 score to evaluate the models on the Tac-Rev, Re-TACRED, and SemEval datasets. On the Tac-Rev and Re-TACRED datasets, we use the large cased version of SpanBERT as the encoder in the model with its default settings. On the SemEval dataset, we use the uncased version of BERT-base as the encoder in the model with its default settings. Stanford CoreNLP Toolkit (SCT) is used for dependency parsing.

C. Baselines

As PLMs have brought many breakthroughs in various NLP tasks in recent years, to evaluate the effectiveness of our model, we compare it with the following powerful baselines:

1) *SMS [12]*: SMS is a novel RE model that employs a hierarchical attention mechanism and global semantic attention to fully exploit multi-granularity features, and then aggregates these extracted features to predict the relation.

2) *SpanBERT [31]*: SpanBERT is a pre-training model that extends BERT using different masking schemes and training objectives. It masks contiguous spans of tokens using a different random process and introduces a span-boundary objective (SBO) that attempts to infer the complete content of the span.

3) *KnowBERT [13]*: To enhance text representations with structured knowledge, the knowledge-enhanced BERT (KnowBERT) incorporates multiple knowledge bases (KBs) into the BERT model and obtains knowledge-enhanced representations that can be used for a variety of downstream tasks.

4) *LUKE [30]*: LUKE is a new pre-trained contextualized representation model. By using a huge entity-annotated corpus, it is trained to predict words and entities that have been randomly masked. With regard to a variety of downstream entity-related tasks, LUKE has demonstrated excellent performance.

5) *GDPNet [39]*: GDPNet creates a multi-view graph to represent various potential relationships among tokens, and the graph is refined through several interactions. Both the refined graph representation and the “[CLS]” token representation of the BERT input sequence is combined to form the input of the softmax classifier, which predicts the type of relation.

6) *TaMM [15]*: The model uses BERT to encode the input, and then incorporates the dependency information by using a type-aware map memory (TaMM) module. TaMM improves relation extraction performance by leveraging dependency type information with an attention mechanism to obtain each dependency's importance.

7) *C-AGGCN [25]*: The model uses dependency trees as inputs and utilizes the graph convolutional network to learn tree structure features in an end-to-end way.

8) *RECENT [40]*: The model introduces mutual restriction of relation and entity type into the relation classification, which can use the entity type to restrict the candidate relations and avoid some unsuitable relations being candidates.

VII. RESULTS AND DISCUSSION

A. Results on Tac-Rev and Re-TACRED

We evaluate our model on the Tac-Rev and Re-TACRED datasets. The experimental results are shown in Table III, and we follow the official train/dev/test split for these two datasets. For our model, feature concatenation is the default feature fusion method. If the gating method introduced in Section V.D is used as the feature fusion method, the model will be denoted as “(with gate)”.

Table III demonstrates that our model yields the highest Macro-F1 scores. When compared to the latest SOTA work such as SMS, the proposed model substantially outperforms the baseline with an absolute improvement of 1.4% on the Tac-Rev dataset and 4.5% on the Re-TACRED dataset. As SMS utilizes origin input features extracted solely from the original input sentences, this proves that our model can benefit from extra knowledge and obtain effective features. Compared to TaMM, we also achieve a 3.2% improvement on the Tac-REV dataset, confirming that comprehensive utilization of origin input features and features extracted from extra knowledge is feasible. It is worth noting that our model outperforms the baselines without the use of an external large knowledge base or large corpus. This also demonstrates the flexibility and effectiveness of our model.

To get a better intuition about how our model works, we conduct an ablation study to analyze the contribution of each component. The results are shown in Table IV. In Table IV, “O” denotes the origin input features, “T” denotes the entity type features, “D” denotes the dependency features, “G” denotes the gating mechanism introduced in Section V.D, and “att” denotes dimension-based attention, for example, SpanBERT+O+T+att means that the model employs SpanBERT as the encoder and utilizes the combination of the origin input features and the entity type features with dimension-based attention. Note that in Table IV, since the computation of the dimensional attention mechanism involves both the original input features and the entity type features, we only add the dimensional attention mechanism when both features are used. Similarly, for the gating mechanism, since its computation involves three feature extractors, we only use it when all three feature extractors are used simultaneously.

As can be seen in Table IV, through the incorporation of various types of features, improvements can be achieved for relation extraction. Specifically, among the three features, we can observe that the entity type feature extractor yields the contribution to the performance with a 1.7% (78.00% vs 79.7%) and 3.1% (85.3% vs 88.4%) improvement on the Tac-Rev and Re-TACRED datasets, respectively. This means that entity type features are important indicators for relation prediction. For the other two types of features, we can also see that they are helpful to improve model performance. As a result, we can see that each of our feature extractors can obtain a boost on the basis of the pre-trained language model, which confirms the feasibility of our feature extractors and shows that all three features are essential for relation extraction tasks.

TABLE III. F1 SCORE RESULTS ON TAC-REV AND RE-TACRED

Models	Tac-Rev (%)	Re-TACRED (%)
SMS [12]	79.8	85.7
SpanBERT [31]	78.0	85.3
KnowBERT [13]	79.3	89.1
LUKE [30]	80.6	-
GDPNet [39]	79.3	-
TaMM [15]	78.0	-
C-AGGCN [25]	75.1	81.0
RECENT [40]	78.7	86.4
Our model	81.2	89.8
Our model (with gate)	**81.2**	**90.2**

TABLE IV. F1 RESULTS OF THE ABLATION STUDY ON THE TAC-REV AND RE-TACRED DATASETS

Models	Tac-Rev (%)	Re-TACRED (%)
SpanBERT	78	85.3
SpanBERT+O	79.5	85.7
SpanBERT+T	79.7	88.4
SpanBERT+D	79.2	88.7
SpanBERT+O+D	81	89.5
SpanBERT+O+T	80.3	88.6
SpanBERT+T+D	80.4	89.4
SpanBERT+O+T+att	80.7	89.7
SpanBERT+O+T+D	80.9	89.7
SpanBERT+O+T+D+att	81.2	89.8
SpanBERT+O+T+D+G	81.1	90
SpanBERT+O+T+D+G+att	**81.2**	**90.2**

We can also observe from Table IV that the model performance can also be gradually improved when the three types of features are combined. For example, on the Tac-Rev dataset, when combining the origin input features and the dependency features, compared to SpanBERT+O and SpanBERT+D, the model SpanBERT+O+D achieves a performance improvement of 1.5% (79.5% vs. 81%) and 1.8% (79.2% vs. 81%), respectively. The results also demonstrate that the combination of the three types of features can bring positive gains.

In summary, the performance of the model is gradually improved with the addition of modules, which confirms that each key component of our model plays a vital role in relation extraction, and deep fusion of origin input features and extra knowledge will further boost the performance of the model. In addition, we can see that using the dimension-based attention mechanism along with the entity type feature can also further leads to performance improvement, which also indicates the effectiveness of the dimension-based attention mechanism.

To further analyze why using features extracted from extra knowledge is effective, the statistical analysis of the relation labels on the Tac-Rev dataset and Re-TACRED dataset is performed, as illustrated in Fig. 5 and 6, respectively.

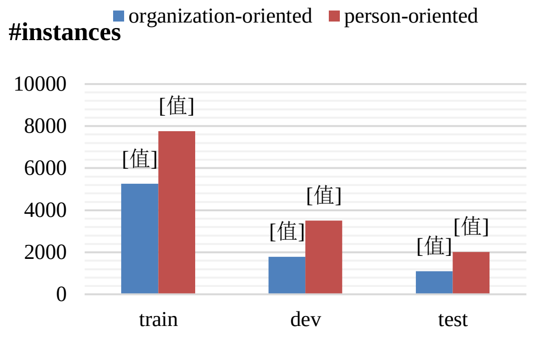


Fig. 5. The statistics of relation labels on the Tac-Rev dataset.

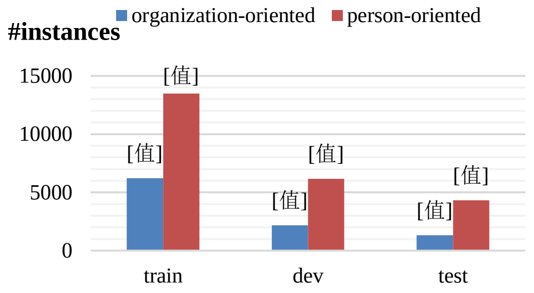


Fig. 6. The statistics of relation labels on the Re-TACRED dataset.

Generally, in these two datasets, each instance is annotated with a person-oriented or organization-oriented relation type, such as per:city_of_birth, per:title, org:employees, and so on, otherwise assigned no_relation for negative instances, and each relation label belongs to the “Person” or “Organization” categories. As the entity types mainly consist of categories such as person, organization, location, and so on. Naturally, inputting the entity type information to the model can play a good hint effect for relation extraction. At the same time, some dependencies also have a significant effect in determining the type of a specific relation. For example, in the sentence “My father built this school”, its' entities are “my father” and “school”, and the dependency between the two entities is obj (Object). When the entity type and dependency information are combined, the model is more likely to predict that the relation type as “org:founded” or “org:founded_by”.

B. *Results on SemEval*

We also conduct experiments on the SemEval dataset. As it is a classical dataset, besides baselines introduced in Section VI.C, the following popular models are adopted as comparison models.

1) LST-AGCN [41]: This model aggregates and transports information about syntactic relations and word features in accordance with the grammatical structure, and directly manipulates the graph to derive the representation for relation classification.

2) DP-GCN [24]: DP-GCN selects relevant information from dependency trees, and each graph convolutional network (GCN) layer contains a selection module that allows it to filter away information that is irrelevant to the target without using any pre-defined rules.

3) C-GCN-MG [26]: The model represents a sentence using multiple sub-graphs and performs graph convolution operations on the sub-graphs to acquire relevant features.

4) C-DAGCN [42]: By appending attention modules over the GCN, C-DAGCN further uses distributional reinforcement to guide the GCN for relational extraction.

5) Two-channel [43]: The model incorporates the benefits of both the Bi-LSTM-ATT and the CNN channel to predict relation.

6) MSML [21]: For the text data, the model constructs feature hierarchy and relation hierarchy and then presents a framework to fully leverage these hierarchies for RE tasks.

7) POS&DP [1]: The model uses both the sequential POS tags and the dependency graph structure for the RE task.

The experimental results are shown in Table V.

TABLE V. F1 RESULTS ON THE SEMEVAL DATASET

Models	SemEval (%)
BERT-base	87.9
SMS [12]	88.3
SpanBERT [31]	-
KnowBERT[13]	89.1
LUKE [30]	-
GDPNet [39]	-
TaMM [15]	89.2
LST-AGCN [41]	86.0
DP-GCN [24]	86.4
C-GCN-MG [26]	85.9
C-DAGCN [42]	86.9
Two-channel [43]	85.42
MSML [21]	89.1
POS&DP [1]	87.2
Our model	**89.4**
Our model (with gate)	89.2

From Table V, we can see that our model also achieves the best result on the SemEval dataset. It outperforms 1.1% over SMS and more than 3% improvement over graph convolution network-based models such as LST-AGCN, DP-GCN, and C-GCN-MG. Compared with the latest works such as POS&DP and MSML, our model still achieves better performance.

The SemEval dataset is not a large dataset, its test set has only 2717 items, and each relation label has fewer instances than the Tac-Rev and Re-TACRED datasets. We conduct ablation experiments on the SemEval dataset to understand the relative contribution of each module of the proposed model. The results shown in Table VI demonstrate that our model can benefit from three types of feature extractors. The meaning of these abbreviations, including “O”, “T”, “D”, “G” and “att”, is the same in Table VI as it is in Table IV. Furthermore, we can see that on the SemEval dataset, the contribution of entity type features to our model is smaller than that of the Tac-Rev and Re-TACRED datasets and the dimension-based attention mechanism makes no impression on the performance. One reason is that for the SemEval dataset, the correlation between relation labels and entity types is not as high as the Tac-Rev and Re-TACRED datasets; Another reason is that entity types that are used to compose features for training the model are generated by entity recognition tool rather than manual annotated, and the noise in the automatically generated entity types may harm the performance of the model; The last reason may be that there are some data belonging to “other” label in the SemEval dataset, which accounts for 17.63% of the training set and 16.71% of the test set, and for these data, the effect of entity type features is limited.

TABLE VI. F1 RESULTS OF THE ABLATION STUDY ON THE SEMEVAL DATASET

Models	SemEval (%)
BERT-base	87.9
BERT+O	88.4
BERT+T	88.2
BERT+D	88.5
BERT+O+D	88.6
BERT+O+T	88.6
BERT+T+D	88.7
BERT+O+T+att	88.6
BERT+O+T+D	89.2
BERT+O+T+D+att	**89.4**
BERT+O+T+D+G	89
BERT+O+T+D+G+att	89.2

C. *Feature Vector Visualizations*

To verify that the dimension-based attention mechanism can enrich the diversity of entity type features, we use t-SNE [44] to project the vector of entity type features into two dimensions. First, we select four sentences from the Re-TACRED dataset, which all contain named entities of type person and organization, and visualize the entity type feature vectors corresponding to the named entities in these sentences, as shown in Fig. 7.

As can be seen from Fig. 7, the four feature vectors belonging to type person are different, and the entity type feature vectors of person type and organization type appear to show two different clusters, i.e., the feature vectors corresponding to two named entities of the same type have a shorter distance on the graph, while the feature vectors corresponding to two named entities of different types have a longer distance on the graph.

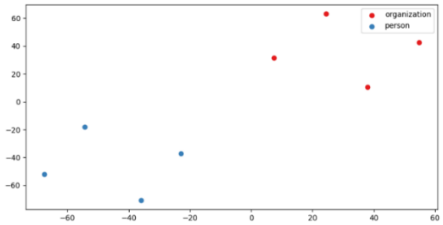


Fig. 7. Visualization of entity type feature vectors from 4 sentences.

Similarly, we select 100 sentences from the Re-TACRED dataset for observation, and the results are shown in Fig. 8.

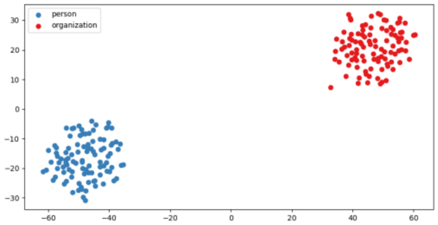


Fig. 8. Visualization of entity type feature vectors from 100 sentences.

From Fig. 8, it can be seen that the feature vectors of type person and organization cluster into two clusters. It indicates that the feature vectors of the same type will be differentiated after applying the dimensional attention mechanism, but the feature vectors of different types can still be clearly distinguished. The visualization results can still meet our expectations when adding more entity types, as shown in Fig. 9. We can see that the dimension-based attention mechanism can effectively increase the discriminability of entity type features, thus solving the problem of the coarse entity types of entity mentions.

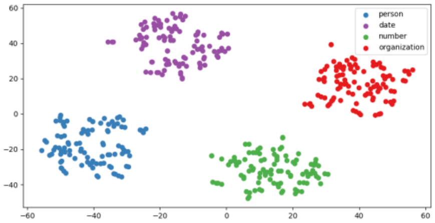


Fig. 9. Visualization of entity type feature vectors from 300 sentences with 4 entity types.

VIII. CONCLUSION

This paper presents a multi-feature fusion model for relation extraction, which explores various information sources and investigates the merging of different types of features in relation extraction. The resulting model can extract abstract features from raw inputs while benefiting from external knowledge. Our study shows that entity type information is especially useful for RE tasks and contributes significantly to the gain in performance from a semantic aspect, while dependency parsing information provides additional benefits. We also demonstrate that deep integration of different types of features makes the proposed model perform significantly better than strong baselines. The experimental results on the three benchmark datasets show that our model is effective and generalizable.

Moreover, our model is mainly composed of a series of feature extractors with a simple architecture. We can add feature extractors according to new external information sources in subsequent research and flexibly integrate them into our model. While our model has achieved good results on annotated datasets, the limitations of its application to unlabeled data still remain for future exploration and resolution. In future work, we will investigate the way to leverage unlabeled data and extend our work to the semi-supervised setting. We also would like to explore knowledge bases to extract additional features and enhance the performance of relation extraction.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 62136002 and 61876027.

REFERENCES

[1] Chen, X., Zhang, M., Xiong, S., and Qian, T. “On the form of parsed sentences for relation extraction,” Knowledge-Based Systems, Volume 251, Article 109184, 2022.

[2] Fan, T., and Wang, H. “Research of Chinese intangible cultural heritage knowledge graph construction and attribute value extraction with graph attention network,” Information Processing & Management, 59(1), Article 102753, 2022.

[3] Zhang, M., Zhou, G., Yu, W., Liu, W. “FAR-ASS: fact-aware reinforced abstractive sentence summarization,” Information Processing & Management, 58(3), Article 102478, 2021.

[4] Yu, M., Yin, W., Hasan, K. S., dos Santos, C., Xiang, B., and Zhou, B. “Improved neural relation detection for knowledge base question answering,” In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 571-581), 2017.

[5] Liu, K. “A survey on neural relation extraction,” Science China Technological Sciences, 2020, 63(10), 1971-1989.

[6] Wang, H., Qin, K., Zakari, R. Y., Lu, G., and Yin, J. “Deep neural network-based relation extraction: an overview,” Neural Computing and Applications, 2022, 34(6), 4781-4801.

[7] Nayak, T., Majumder, N., Goyal, P., and Poria, S. “Deep neural approaches to relation triplets extraction: a comprehensive survey,” Cognitive Computation, 2021, 13(5), 1215-1232.

[8] Zeng, D., Liu, K., Lai, S., Zhou, G., and Zhao, J. “Relation classification via convolutional deep neural network,” In Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers (pp. 2335-2344), 2014.

[9] Zhang, R., Meng, F., Zhou, Y., and Liu, B. “Relation classification via recurrent neural network with attention and tensor layers,” Big Data Mining and Analytics, 2018, 1(3), 234-244.

[10] Hashimoto, K., Miwa, M., Tsuruoka, Y., and Chikayama, T. “Simple customization of recursive neural networks for semantic relation classification,” In Proceedings of the 2013 conference on empirical methods in natural language processing (pp. 1372-1376), 2013.

[11] Liu, J., Chen, S., Wang, B., Zhang, J., Li, N., and Xu, T. “Attention as relation: learning supervised multi-head self-attention for relation

extraction," In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence (pp. 3787-3793), 2021.

[12] Liang, X., Wu, S., Li, M., and Li, Z. "Modeling multi-granularity hierarchical features for relation extraction," In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL) (pp. 5088–5098), 2022.

[13] Peters, M. E., Neumann, M., Logan, R., Schwartz, R., Joshi, V., et al. "Knowledge enhanced contextual word representations," 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) (pp. 43-54), 2019.

[14] Wang, R., Tang, D., Duan, N., Wei, Z., Huang, X. J., et al. "K-Adapter: infusing knowledge into pre-trained models with adapters," In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021 (pp. 1405-1418), 2021.

[15] Chen, G., Tian, Y., Song, Y., and Wan, X. "Relation extraction with type-aware map memories of word dependencies," In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021 (pp. 2501-2512), 2021.

[16] Vashishth, S., Joshi, R., Prayaga, S. S., Bhattacharyya, C., and Talukdar, P. "RESIDE: improving distantly-supervised neural relation extraction using side information," In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (pp. 1257-1266), 2018.

[17] Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Séaghdha, D. Ó., et al. "SemEval-2010 Task 8: multi-way classification of semantic relations between pairs of nominals," In Proceedings of the 5th International Workshop on Semantic Evaluation (pp. 33-38), 2010.

[18] Du, L., Kumar, A., Johnson, M., and Ciaramita, M. "Using entity information from a knowledge base to improve relation extraction," In Proceedings of the Australasian Language Technology Association Workshop 2015 (pp. 31-38), 2015.

[19] Kambhatla, N. "Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction," In Proceedings of the ACL interactive poster and demonstration sessions (pp. 178-181), 2004.

[20] Zhou, G., Su, J., Zhang, J., and Zhang, M. "Exploring various knowledge in relation extraction," In Proceedings of the 43rd annual meeting of the association for computational linguistics (ACL'05) (pp. 427-434), 2005.

[21] Zhang, M., Qian, T., and Liu, B. "Exploit feature and relation hierarchy for relation extraction," IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2022, 30, 917-930.

[22] Nguyen, T. H., and Grishman, R. "Relation extraction: perspective from convolutional neural networks," In Proceedings of the 1st workshop on vector space modeling for natural language processing (pp. 39-48), 2015.

[23] Wen, H., Zhu, X., Zhang, L., Li, F." A gated piecewise CNN with entity-aware enhancement for distantly supervised relation extraction, " Information Processing & Management, 57(6), Article 102373, 2020.

[24] Yu, B., Mengge, X., Zhang, Z., Liu, T., Yubin, W., et al. "Learning to prune dependency trees with rethinking for neural relation extraction，" In Proceedings of the 28th International Conference on Computational Linguistics (pp. 3842-3852)， 2020.

[25] Guo, Z., Zhang, Y., and Lu, W. "Attention guided graph convolutional networks for relation extraction," In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 241-251), 2019.

[26] Mandya, A., Bollegala, D., and Coenen, F. "Graph convolution over multiple dependency sub-graphs for relation extraction," In Proceedings of the 28th International Conference on Computational Linguistics (COLING) (pp. 6424-6435), 2020.

[27] Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. "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) (pp. 4171-4186), 2019.

[28] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdino, R., et al. "XLNet: generalized autoregressive pretraining for language understanding," In Proceedings of the 33rd International Conference on Neural Information Processing Systems (NIPS) (pp. 5753-5763), 2019.

[29] Hou, J., Li, X., Yao, H., Sun, H., Mai, T.,et al. "Bert-based Chinese relation extraction for public security," IEEE Access, 2020, 8, 132367-132375.

[30] Yamada, I., Asai, A., Shindo, H., Takeda, H., and Matsumoto, Y. "LUKE: deep contextualized entity representations with entity-aware self-attention," In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 6442-6454), 2020.

[31] Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., et al. "SpanBERT: improving pre-training by representing and predicting spans," Transactions of the Association for Computational Linguistics, 2020, 8, 64-77.

[32] Wang, X., Gao, T., Zhu, Z., Zhang, Z., Liu, Z., et al. "KEPLER: a unified model for knowledge embedding and pre-trained language representation." Transactions of the Association for Computational Linguistics, 2021, 9, 176-194.

[33] Miller, A., Fisch, A., Dodge, J., Karimi, A. H., Bordes, A., et al. "Key-value memory networks for directly reading documents," In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (pp. 1400-1409), 2016.

[34] Song, Y., Tian, Y., Wang, N., and Xia, F. "Summarizing medical conversations via identifying important utterances," In Proceedings of the 28th International Conference on Computational Linguistics (pp. 717-729), 2020.

[35] Tian, Y., Chen, G., and Song, Y. "Enhancing aspect-level sentiment analysis with word dependencies," In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume (pp. 3726-3739), 2021.

[36] Alt, C., Gabryszak, A., and Hennig, L. "TACRED Revisited: a thorough evaluation of the TACRED relation extraction task," In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 1558-1569), 2020.

[37] Stoica, G., Platanios, E. A., and Póczos, B. "Re-TACRED: addressing shortcomings of the TACRED dataset," In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 15, pp. 13843-13850), 2021.

[38] Zhang, Y., Zhong, V., Chen, D., Angeli, G., and Manning, C. D. "Position-aware attention and supervised data improve slot filling," In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (pp. 35-45), 2017.

[39] Xue, F., Sun, A., Zhang, H., and Chng, E. S. "GDPNet: refining latent multi-view graph for relation extraction," In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 16, pp. 14194-14202), 2021.

[40] Lyu, S., and Chen, H. "Relation Classification with Entity Type Restriction," In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021 (pp. 390-395), 2021.

[41] Sun, K., Zhang, R., Mao, Y., Mensah, S., and Liu, X. "Relation extraction with convolutional network over learnable syntax-transport graph," In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 05, pp. 8928-8935), 2020.

[42] Li, Z., Sun, Y., Zhu, J., Tang, S., Zhang, C., and Ma, H. "Improve relation extraction with dual attention-guided graph convolutional networks," Neural Computing and Applications, 2021, 33(6), 1773-1784.

[43] Wang, Y., Han, Z., You, K., Lin, Z. "A Two-channel model for relation extraction using multiple trained word embeddings," Knowledge-Based Systems, Volume 255, Article 109701, 2022.

[44] Maaten, L. and Hinton, G. "Visualizing data using t-SNE." Journal of machine learning research, 2008 ,9(86):2579-2605.