Modeling Multi-Granularity Hierarchical Features for Relation Extraction

Xinnian Liang[1]*, Shuangzhi Wu[2], Mu Li[2] and Zhoujun Li[1]†
[1]State Key Lab of Software Development Environment, Beihang University, Beijing, China
[2]Tencent Cloud Xiaowei, Beijing, China
{xnliang,lizj}@buaa.edu.cn; frostwu@tencent.com,limugx@qq.com;

Abstract

Relation extraction is a key task in Natural Language Processing (NLP), which aims to extract relations between entity pairs from given texts. Recently, relation extraction (RE) has achieved remarkable progress with the development of deep neural networks. Most existing research focuses on constructing explicit structured features using external knowledge such as knowledge graph and dependency tree. In this paper, we propose a novel method to extract multi-granularity features based solely on the original input sentences. We show that effective structured features can be attained even without external knowledge. Three kinds of features based on the input sentences are fully exploited, which are in entity mention level, segment level, and sentence level. All the three are jointly and hierarchically modeled. We evaluate our method on three public benchmarks: SemEval 2010 Task 8, Tacred, and Tacred Revisited. To verify the effectiveness, we apply our method to different encoders such as LSTM and BERT. Experimental results show that our method significantly outperforms existing state-of-the-art models that even use external knowledge. Extensive analyses demonstrate that the performance of our model is contributed by the capture of multi-granularity features and the model of their hierarchical structure. Code and data are available at https://github.com/xnliang98/sms.

1 Introduction

Relation extraction (RE) is a fundamental task in Natural Language Processing (NLP), which aims to extract relations between entity pairs from given plain texts. RE is the cornerstone of many downstream NLP tasks, such as knowledge base construction (Ji and Grishman, 2011), question answering (Yu et al., 2017), and information extraction (Fader et al., 2011).

Most recent works focus on constructing explicit structured features using external knowledge such as knowledge graph, entity features and dependency tree. To infuse prior knowledge from existing knowledge graph, recent works (Peters et al., 2019a; Wang et al., 2020b,a) proposed some pretrain tasks to help model learn and select proper prior knowledge in the pre-training stage. Baldini Soares et al. (2019); Yamada et al. (2020); Peng et al. (2020) force model learning entitiy-related information via well-designed pre-train tasks. Zhang et al. (2018); Guo et al. (2019); Xue et al. (2020); Chen et al. (2020) encode dependency tree with graph neural network (Kipf and Welling, 2017) (GNN) to help RE models capture non-local syntactic relation. All of them achieve a remarkable performance via employing external information from different structured features.

However, they either need time-consuming pretraining with external knowledge or need an external tool to get a dependency tree which may introduce unnecessary noise. In this paper, we aim to attain effective structured features based solely on the original input sentences. To this end, we analyze previous typical works and find that three kinds of features mainly affect the performance of RE models, which are entity mention level[1], segment[2] level and sentence level features. Sentence level and entity mention (Baldini Soares et al., 2019; Yamada et al., 2020; Peng et al., 2020) level features were widely used by previous works but segment level feature (Yu et al., 2019; Joshi et al., 2020) does not get as much attention as the previous two features. These three level features can provide different granularity information from input sentences for relation prediction (Chowdhury and Lavelli, 2012; Kim). However, recent works did not consider them at the same time and ignored

*Contribution during internship at Tencent Inc.

†Corresponding Author

[1] entity mentions contain the entity itself and co-references of it.

[2] continuous words in sentence (n-gram)

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5088 - 5098
July 10-15, 2022 ©2022 Association for Computational Linguistics

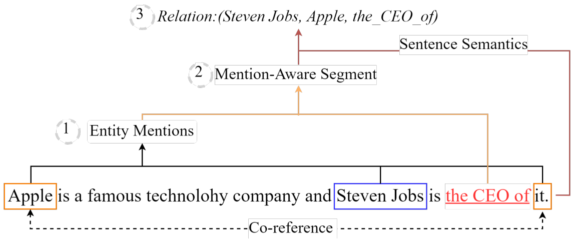

3 Relation:(Steven Jobs, Apple, the_CEO_of)
Sentence Semantics
2 Mention-Aware Segment
1 Entity Mentions
Apple is a famous technolohy company and Steven Jobs is the CEO of it.
Co-reference

Figure 1: An Example for relation extraction which shows the hierarchical structure between entity mention level and segment level features.

the structure and interactive of them.

We employ a simple example in Figure 1 to show the hierarchical and joint structure of the previous three granularities features. The hierarchical structure is between mention level and segment level features. This example gives sentence and entity pairs (“*Steven Jobs*", “*Apple*"). We can find that the relation “*the_CEO_of*" of given entity “*Apple*" and entity “*Steven Jobs*" is built upon the core segment “*the CEO of*" between the entity mention “*it*" (i.e. co-reference of entity “*Apple*") and entity “*Steven Jobs*". To extract relation from this example, RE models need to first capture mention level features of given entities and then catch core segment level feature “*the CEO of*" which is related to mention level features. Finally, RE models can easily predict the relation with previous two-level hierarchical features. Besides, sentence-level semantic features can assist RE models to predict the relation of examples without an explicit pattern of entity mentions and segments.

Following previous intuitive process, we propose a novel method which extracts multi-granularity features based solely on the original input sentences. Specifically, we design a hierarchical mention-aware segment attention, which employs a hierarchical attention mechanism to build association between entity mention level and segment level features. Besides, we employ a global semantic attention to get a deeper understanding of sentence level features from input sentence representation. Finally, we aggregate previously extracted multi-granularity features with a simple fully-connected layer to predict the relation.

To evaluate the effectiveness of our method, we combine our method with different text encoders (e.g. LSTM and BERT) and results show that our method can bring significant improvement for all of them. Compared with models without using external knowledge, SpanBERT with our method can achieve a new state-of-the-art result on Semeval 2010 Task 8, Tacred and Tacred Revisited. It is deserved to mention that the performance of our model is very competitive with the state-of-the-art models, which employ large-scale extra training data or information. We also do many analyses to demonstrate that features from representation of input itself are enough for the sentence-level RE tasks and multi-granularity features with hierarchical structure are crucial for relation prediction.

2 Methodology

The structure of our model and details of each component is shown in figure 2. We can see the overall architecture in the middle. It is divided into three components from bottom to top: 1) A text encoder which is employed to obtain text vector representations; 2) A multi-granularity hierarchical feature extractor which can exploit effective structured features from text representations; 3) A feature aggregation layer which aggregate previous multi-granularity features for relation prediction. In this section, we will introduce details of three components.

Firstly, we formalize the relation extraction task. Let $\mathtt{x} = \{x_1, x_2, ..., x_n\}$ be a sequence of input tokens, where $x_0 = $ [CLS] and $x_n = $ [SEP] are special start and end tokens for BERT-related encoders. Let $s_1 = (i, j)$ and $s_2 = (k, l)$ be pairs of entity indices. The indices in s_1 and s_2 delimit entities in x: $[x_i, \dots, x_{j-1}]$ and $[x_k, \dots, x_{l-1}]$. Our goal is to learn a function $P(r) = f_\theta(\mathtt{x}, s_1, s_2)$, where $r \in \mathcal{R}$ indicates the relation between the entity pairs, which is marked by $s1$ and $s2$. $\mathcal{R}$ is a pre-defined relation set.

2.1 Encoder Layer

We first employ a text encoder (e.g. BERT) to map tokens in input sentences into vector representations which can be formalized by Equ. (1).

$$\mathbf{H} = \{h_0, \dots, h_n\} = f_{encoder}(x_0, \dots, x_n) \quad (1)$$

Where $\mathbf{H} = \{h_0, \dots, h_n\}$ is the vector representation of input sentences.

Our work is built upon $\mathbf{H}$ and does not need any external information. We employ a max-pooling operation to obtain shallow features of entity pairs and input sentences. $h_{e_1} = \mathtt{Maxpooling}(h_{i:j})$ and $h_{e_2} = \mathtt{Maxpooling}(h_{k:l})$ are the representations of entity pairs. $h_g = \mathtt{Maxpooling}(\mathbf{H})$ is the vector representation of input sentences which contains global semantic information.

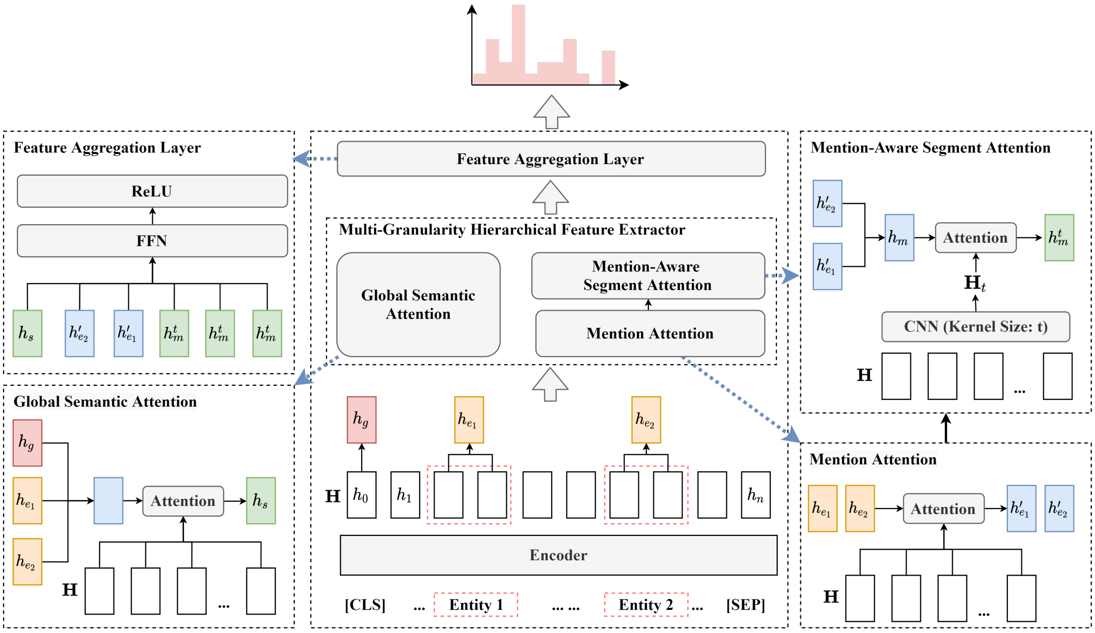


Figure 2: **Middle:** The structure of our proposed multi-granularity hierarchical feature extractor. **Left:** Details of global semantic attention (sentence level feature) and feature aggregation layer. **Right:** Details of mention attention (entity mention level feature) and mention-aware segment attention (segment level feature).

2.2 Multi-Granularity Hierarchical Feature Extractor

The multi-granularity hierarchical feature extractor is the core component of our method and it consists of three attention mechanism for different granularity features extraction: 1) mention attention which is designed to entity mention features of given entity pairs; 2) mention-aware segment attention which is based on the entity mention features from previous mention attention and aim to extract core segment level feature which is related to entity mentions; 3) global semantic attention which focuses on the sentence level feature.

2.2.1 Mention Attention

The structure of mention attention is shown in the right bottom of Figure 2. To capture more information about given entity pairs from input sentences, we extract entity mention level features by modeling the co-references (mentions) of entities implicitly. We employ a mention attention to capture information about entity 1 and 2 respectively. Specifically, we can use the representation of an entity as a query to obtain the entity mention feature from $\mathbf{H}$ by Equ. (2).

$$\begin{aligned} h'_{e_1} &= \texttt{Softmax}(\frac{\mathbf{H} \cdot h_{e_1}}{\sqrt{d}}) \cdot \mathbf{H} \\ h'_{e_2} &= \texttt{Softmax}(\frac{\mathbf{H} \cdot h_{e_2}}{\sqrt{d}}) \cdot \mathbf{H} \end{aligned} \qquad (2)$$

Where d is the dimension of vector representation and used to normalize vectors. Then, h'_{e_1} and h'_{e_2} model the mentions of given entity pairs implicitly and contain more entity semantic information than h_{e_1} and h_{e_2}.

2.2.2 Mention-Aware Segment Attention

The structure of mention-aware segment attention is shown in the right top of Figure 2. And the mention-aware segment attention is a hierarchical structure based on the entity mention features h'_{e_1} and h'_{e_2} from mention attention.

Before introducing mention-aware segments attention, we first introduce how to get the representations of segments. We employ convolutional neural networks (CNN) with different kernel sizes to obtain all n-gram segments in texts, which can effectively capture local n-gram information with Equ. (3).

$$\mathbf{H}_t = \texttt{CNN}_t(\mathbf{H}), t \in \{1, 2, 3\} \qquad (3)$$

Where t is the kernel size of CNN and is empirically set as $t \in \{1, 2, 3\}$ which means extract 1-gram, 2-gram ,and 3-gram segment level features.

Intuitively, the valuable segments should be highly related to given entity pairs, which can help the model to decide the relation of given entity pairs. Entity mention features h'_{e_1} and h'_{e_2} contain comprehensive information of given entity pairs and $\mathbf{H}_t$ contain 1,2,3-gram segment level features. We can extract mention-aware segment level features by simply linking them with attention mechanisms by Equ. (4).

$$h_m^t = \texttt{Softmax}(\frac{\mathbf{H}_t \cdot (W_m[h'_{e_1}; h'_{e_2}])}{\sqrt{d}}) \cdot \mathbf{H}_t \quad (4)$$

Then, we get $\{h_m^t\}_{t=1,2,3}$ which capture different granularity segments features.

2.2.3 Global Semantic Attention

The structure of global semantic attention is shown in the left bottom of Figure 2. Previous works always directly concatenate vector representation $[h_{e_1}; h_{e_2}; h_g]$ as the global semantic feature of input text. We argue this is not enough to help model capture deeper sentence level semantic information for RE. Different from them, to obtain better global sentence-level semantic feature, we employ an attention operation called global semantic attention which use the concatenation of $[h_{e_1}; h_{e_2}; h_g]$ as query to capture deeper semantic feature from context representation $\mathbf{H}$ by Equ. (5).

$$h_s = \texttt{Softmax}(\frac{\mathbf{H} \cdot (W_s[h_{e_1}; h_{e_2}; h_g])}{\sqrt{d}}) \cdot \mathbf{H} \quad (5)$$

Where $W_s \in R^{d\times 3d}$ is a linear transform matrix, and d is a hidden dimension of vectors. The concatenation of $[h_{e_1}; h_{e_2}; h_g]$ is used as a query of the attention operation, which can force the extracted global semantic representation h_s contain entity mention related sentence level feature.

2.3 Feature Aggregation Layer

The structure of the feature aggregation layer is shown in the left top of Figure 2. We aggregate previous multi-granularity features by Equ. (6).

$$h_o = \texttt{ReLU}(W_a[h_s; h'_{e_1}; h'_{e_1}; h_m^1;; h_m^2;; h_m^3]) \quad (6)$$

Where $W_a \in R^{6d\times d}$ is a linear transform matrix and ReLU is a nonlinear activation function.

2.4 Classification

Finally, we employ a softmax function to output the probability of each relation label as follows:

$$P(r|\mathrm{x}, s_1, s_2) = \texttt{Softmax}(W_o h_o) \quad (7)$$

The whole model is trained with cross entropy loss function. We call the multi-granularity hierarchical feature extractor: SMS (relation extraction with **S**entence level, **M**ention level and mention-aware **S**egment level features).

	Tacred	Semeval
lr	3e-5	2e-5
warmup steps	300	0
batch size	64	32
V100 GPU	4x	1x
epochs	4	10
max length	128	128

Table 1: Hyper-parameters used in training.

3 Experiments

3.1 Datasets

We evaluate the performance of our method on Semeval 2010 Task 8, Tacred and Tacred Revisited datasets.

SemEval 2010 Task 8 (Hendrickx et al., 2010) is a public dataset which contains 10,717 instances with 9 relations. The training/validation/test set contains 7,000/1,000/2,717 instances respectively.

Tacred[3] is one of the largest, most widely used crowd-sourced datasets for Relation Extraction (RE), which is introduced by (Zhang et al., 2017), with 106,264 examples built over newswire and web text from the corpus used in the yearly TAC Knowledge Base Population (TAC KBP) challenges. The training/validation/test set contains 68,124/22,631/15,509 instances respectively. It covers 42 relation types including 41 relation types and a *no_relation* type and contains longer sentences with an average sentence length of 36.4.

Tacred Revisited[4] was proposed by (Alt et al., 2020) which aims to improve the accuracy and reliability of future RE method evaluations. They validate the most challenging 5K examples in the development and test sets using trained annotators and find that label errors account for 8% absolute F1 test error, and that more than 50% of the examples need to be relabeled. Then, they relabeled the test set and released the Tacred Revisited dataset.

3.2 Settings

The setting of hyper-parameters is shown in table 1. Following the implementation details mentioned in (Zhang et al., 2017), we employ the "entity mask" strategy and the "multi-channel" strategy during experiments. The former means replacing each subject entity (and object entity similarly) in the original sentence with a special `[SUBJ-⟨NER⟩]` token. All of our reported results are the mean of 5 results with different seeds, which are ran-

[3] https://nlp.stanford.edu/projects/tacred/

[4] https://github.com/DFKI-NLP/tacrev

	Model		Tacred			Tacred Revisited	
		P(ΔP)	R(ΔR)	F1($\Delta F1$)	P(ΔP)	R(ΔR)	F1($\Delta F1$)
1	LSTM	62.5	63.4	62.9	67.7	**73.1**	70.3
	PA-LSTM*	65.7	64.5	65.1	73.6	72.8	73.2
	SA-LSTM	68.1	**65.7**	66.9	**78.3**	72.5	**75.4**
	C-GCN*	69.9	63.3	66.4	76.8	71.4	74.1
	C-AGGCN*	**71.9**	63.4	**67.5**	78.2	70.5	74.3
2	TRE	-	-	67.4	-	-	75.3
	BERT-base	68.1	67.7	67.9	69.4	75.8	72.6
	BERT-large	69.2	69.4	69.3	75.1	74.8	75.0
	BERT+LSTM	73.3	63.1	67.8	74.1	73.9	74.0
	SpanBERT-base	67.6	68.6	68.1	69.1	78.2	73.7
	SpanBERT-large	70.8	70.9	70.8	77.8	78.3	78.0
	DG-SpanBERT-large*	**71.4**	**71.6**	**71.5**	**79.2**	**78.6**	**78.9**
3	MTB†	-	-	71.5	-	-	-
	KnowBERT-W+W‡	**71.6**	71.4	71.5	79.0	79.6	79.3
	KEPLER‡	70.4	73.0	71.7	-	-	-
	K-Adapter‡	70.1	74.0	72.0	-	-	-
	LUKE†	70.4	**75.1**	**72.7**	**79.7**	**80.6**	**80.2**
4	LSTM+SMS	**72.8(+10.3)**	64.6(1.2)	68.4(+5.5)	80.8(+3.1)	72.2(-0.9)	75.9(+5.6)
	SpanBERT-base+SMS	72.6(+5.0)	68.4(-0.2)	70.5(+2.4)	79.1(+10.0)	77.7(-0.5)	78.3(+4.6)
	SpanBERT-large+SMS	72.2(+1.4)	**71.6(+0.7)**	**71.9(+1.1)**	**79.3(+1.5)**	**80.4(+2.1)**	**79.8(+1.8)**
	BERT-base+SMS	69.4(+1.3)	70.2(+2.5)	69.7(+1.8)	77.0(+7.6)	80.1(+4.3)	78.5(+5.9)
	BERT-large+SMS	70.7(+1.5)	69.1(-0.3)	69.9(+0.6)	78.9(+3.8)	79.2(+4.4)	79.1(+4.1)

Table 2: Results on Tacred and Tacred Revisited. Bold means the best results in each block. Underline means the best results in block 1, 2, and 4. * means that the model employs dependency tree information. †means that the model employs extra training data to pre-train the model. ‡means the model employs knowledge graphs.

domly selected. We evaluate the models on Tacred with the official script[5] in terms of the Macro-F1 score and on Semeval with the official script *semeval2010_task8_scorer-v1.2.pl.*

When employing LSTM as the encoder, we employ a single-layer bidirectional LSTM with the hidden dimension size set to 200, we set dropout after the input layer and before the output layer with $p = 0.5$. We use stochastic gradient descent (SGD) with epochs of 30, learning rate of 1.0, decay weight of 0.5 and batch sizes of 50 to train the model. The latter is to augment the input by concatenating it with part-of-speech (POS) and named entity recognition (NER) embeddings. Glove (Pennington et al., 2014) embedding with 300-dimension is used for initializing word embedding layers in LSTM+SMS. NER embedding, POS embedding and position embedding are randomly initialized with 30-dimension vectors from uniform distribution.

3.3 Comparison Models

We mainly compare with models which are based on pre-trained language models (e.g. BERT). We reproduce the results of **BERT** and **SpanBERT** to evaluate the improvement of our method. We also compared other models with pre-trained language models. **TRE** (Alt et al., 2019), which uses the unidirectional OpenAI Generative Pre-Trained Transformer (GPT) (Radford et al., 2019). **BERT-LSTM** (Shi and Lin, 2019), which stacks bidirectional LSTM layer to BERT encoder. **DG-SpanBERT**, which replaced the encoder of C-GCN (Zhang et al., 2018) with SpanBERT and achieved the new state-of-the-art result without extra training data. **MTB** (Baldini Soares et al., 2019), which incorporates an intermediate "matching the blanks" pre-training on the entity-linked text based on English Wikipedia. **KnowBERT-W+W** (Peters et al., 2019b), which is an an advanced version of KnowBERT. **KEPLER** (Wang et al., 2020b), which integrates factual knowledge with the supervision of the knowledge embedding objective. **K-Adapter** (Wang et al., 2020a), which consists of a RoBERTa model and an adapter to select adaptive knowledge. **LUKE** (Yamada et al., 2020), which is trained with a new pre-training task which involves predicting randomly masked words and entities in a large entity-annotated corpus retrieved from Wikipedia and introduce a new entity-aware attention mechanism.

In order to further prove the effectiveness of our SMS, we use bi-directional LSTM as encoder, and compare with models which do not use pre-trained language models. We choose two sequence-based models. **PA-LSTM** (Zhang et al., 2017), which

[5] https://github.com/yuhaozhang/tacred-relation

employ Bi-LSTM to encoder the plain text and combine with position-aware attention mechanism to extract relation. PA-SLTM is the benchmark of Tacred. **SA-LSTM** (Yu et al., 2019), which employ CRF to learn segment-level attention and is the best sequence-based model of Tacred.

We also compare our model with two other dependency-based models which make use of GCN (Kipf and Welling, 2017) to capture semantic information from the dependency tree. **C-GCN** (Zhang et al., 2018), which applies pruning strategy and GCN to extract features from tree structure for relation extraction. **C-AGGCN** (Guo et al., 2019), which introduces self-attention to build a soft adjacent matrix as input of Dense GCN to learn tree structure features.

3.4 Results on Tacred and Tacred Revisited

We first report the results or our model on Tacred and Tacred Revisited on Table 2. Compared models are divided into three categories: 1) models with Bi-LSTM encoder in block 1; 2) models with pre-trained models in block 2; 3) models with external knowledge in block 3. The results of our model are reported in block 4. We use * to mark models with dependency trees which are obtained with external tools. We use †to mark models which use external training data to pre-train the model and ‡to mark models which employ knowledge graphs to pre-train or fine-tune the model. Models with †and ‡require external data and we do not directly compare them.

3.4.1 Compare with Pre-trained models

We can see that our SMS can bring at least 0.6 and up to 5.5 F1 score improvement for the original encoder on Tacred dataset. On the Tacred Revisited dataset, our SMS can bring at least 1.8 and up to 5.9 F1 score improvement for the original encoder. Overall, different encoders with SMS all can obtain remarkable improvement on both datasets. This proves that our SMS really captures effective features from input sentence representations, which can not get directly from the representations. Compared with models which employ pre-trained models without external knowledge (i.e. training data or knowledge graph) in block 2, pretrained models with our SMS in block 4 overall perform better and SpanBERT-large+SMS achieve new state-of-the art results on both datasets. In addition, we can see that the performance of SpanBERT-large+SMS is better than MTB, KnowBERT-W+W, and KEPLER in block 3 and is competitive with K-Adapter and LUKE.

Models	SemEval
	F1($\Delta F1$)
LSTM	82.7
LSTM+Attention	84.0
TRE	87.1
BERT-base	87.9
BERT-large	88.8
SpanBERT-base	88.2
SpanBERT-large	89.4
R-BERT (Wu and He, 2019a)	89.3
MTB †	89.5
KnowBert-W+W ‡	89.1
LSTM+SMS	86.8(+4.1)
BERT-base+SMS	88.4(+0.5)
BERT-large+SMS	89.8(+1.0)
SpanBERT-base+SMS	88.5(+0.3)
SpanBERT-large+SMS	**90.3(+0.9)**

Table 3: Results on SemEval 2010 task8. †means that the model employs external knowledge to pre-train the model. ‡means the model employs a knowledge base.

The increase of F1 score is more conspicuous on Tacred Revisited compared with Tacred. This phenomenon is further evidence that existing models have neared the upper limit of Tacred, which have many mislabeled examples. Besides, we can see that models based on SpanBERT all have a pretty good performance. This phenomenon proves the importance of segment level features.

3.4.2 Compare with LSTM-based models

To further evaluate the effectiveness of our method SMS, we specially combine SMS with LSTM encoder. We can observe that our model also outperforms the model with LSTM encoder in block 1. Dependency-based models with graph neural networks (C-GCN and C-AGGCN) have a remarkable performance on Tacred and models which focus on segments (SA-LSTM) have a better performance on Tacred Revisited. This phenomenon means that directly modeling the segment level feature can not effectively overcome the noise from mislabeled examples and the introduction of graph structure with dependency trees can help models tackle some influence from wrong examples in the dataset itself.

However, our LSTM+SMS can outperform them on both datasets due to our mention-aware segment attention can alleviate influence from mislabeled entity pairs via modeling entity mention level feature and hierarchical structure.

	F1($\Delta F1$)
SpanBERT-large	78.0
+ sentence level	78.6(+0.6)
+ mention level	78.8(+0.8)
+ segment level	79.4(+1.4)
+ all	**79.8(+1.8)**

Table 4: Ablation study on Tacred Revisited test set.

3.5 Results on SemEval 2010 Task 8

We also evaluate our SMS with different encoders on SemEval 2010 Task 8 dataset and results are reported in Tab. 3. We can observe that our SMS still brings remarkable improvement for different encoders, especially for LSTM encoders. SpanBERT-large+SMS outperforms all compared to strong baselines. Besides, SpanBERT-large+SMS can beat models with external knowledge due to this dataset being simpler than Tacred which only has 9 relations and shorter input sentences. These reasons reduce the gain from the introduction of external knowledge.

We also can see that the improvement of LSTM with SMS is up to 4.1% F1 score. We guess that pre-trained models contain a lot of semantic information from pre-training data which is similar to features from our SMS. However, LSTM only captures features from the plain texts and can achieve more improvement from our proposed SMS.

4 Discussion

4.1 Ablation Study

To evaluate the contribution of each component of our SMS, we do an ablation study and results are shown in Tab. 4. We can observe that segment level features contribute the most for the F1 score, which are extracted by the mention-aware segment attention. This means the hierarchical structure between entity mention level and segment level feature really play a vital role for relation prediction. In the future works, segment features need more attention. We also can see that all three granularity features influence the performance obviously. This proves the capture of these three granularity features are proper for relation extraction tasks.

4.2 Analysis with N-gram Segments

We show the performance on the Tacred Revisited test set with different n-gram segments features in Figure 3. Number n in the x-axis means the model uses $1-n$-gram segment features. We can observe that the model with 1,2,3-gram segment features achieves the best performance. Longer segment features can not bring improvement and may bring noise to the performance of the model. So we employ 1,2,3-gram segment level features in our paper.

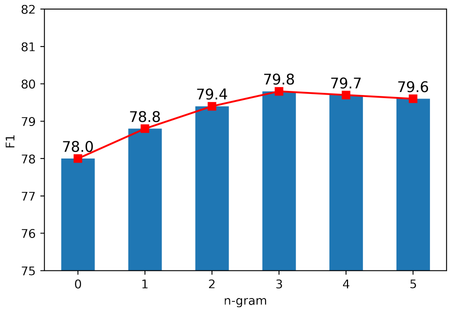


Figure 3: Performance on Tacred Revisited test set with different n-gram segments features. Number n in x-axis means the model use n-gram segment features.

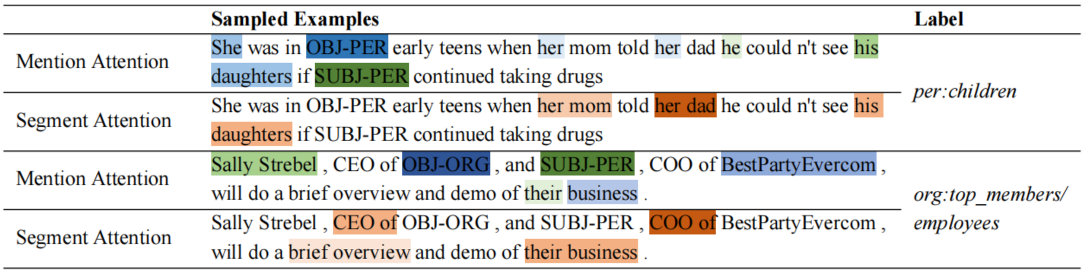
4.3 Case Study

As shown in Figure 4, we visualized the attention of our SMS with two examples which are sampled from Tacred test set. In the first example, our method successfully pays more attention to entity mentions: "*she*", "*her*", "*he*", and "*his*". All of them are key entity mentions for the predicted relation. We also can observe that the mention-aware segment attention of our SMS can focus on the core segment "*her dad*", which is highly related to given entity pairs and matches the predicted label "*per:children*". From the second example, we can see that the model learns additional information which is similar to target relation. The model not only successfully pays attention on entity mention "*SUBJ-PER*" and core segment "*COO of*", but also captures similar entity mention "*Sally Strebel*" and segment "*CEO of*" simultaneously. The case study proves that the mention attention and mention-aware segment attention do capture crucial entity mention level and segment level features.

5 Related Works

5.1 RE with Neural Networks

In recent years, neural networks have been large-scale used in relation extraction (RE). Zeng et al. (2014); Nguyen and Grishman (2015); Wang et al. (2016) employ convolutional neural networks (CNN) to extract lexical and sentence level features for RE. Zhang and Wang (2015) employs



	Sampled Examples	Label
Mention Attention	She was in OBJ-PER early teens when her mom told her dad he could n't see his daughters if SUBJ-PER continued taking drugs	*per:children*
Segment Attention	She was in OBJ-PER early teens when her mom told her dad he could n't see his daughters if SUBJ-PER continued taking drugs	
Mention Attention	Sally Strebel , CEO of OBJ-ORG , and SUBJ-PER , COO of BestPartyEvercom , will do a brief overview and demo of their business .	*org:top_members/ employees*
Segment Attention	Sally Strebel , CEO of OBJ-ORG , and SUBJ-PER , COO of BestPartyEvercom , will do a brief overview and demo of their business .	

Figure 4: Two examples which are sampled from Tacred Revisited test set. The shade of the color represents how much attention is allocated. For the sake of perception, we did not color words with very low attention values.

bidirectional recurrent neural networks (RNN) to learn long-term features to tackle long-term relation problems in RE. And many models with different attention mechanisms were proposed (Zhou et al., 2016; Zhang et al., 2017; Xiao and Liu, 2016; Wang et al., 2016; Yu et al., 2019). Vu et al. (2016); Nayak and Ng (2019) combine CNN and RNN to extract multi-types features from input sentences. Recently, Verga et al. (2018); Liu et al. (2020) employ new neural structure transformer to extract features for RE, which is based on self-attention and is robust and powerful.

Different from previous sequence-based models, dependency-based models employ dependency parsing of input sentences to capture non-local syntactic relations. The use of dependency trees has been a trend in relation extraction (Xu et al., 2015; Cai et al., 2016; Miwa and Bansal, 2016; Song et al., 2018). Peng et al. (2017) split the dependency graph into two directed graphs, then extended the tree LSTM model (Tai et al., 2015) based on these two graphs to learn the structure of syntax dependency. Zhang et al. (2018) first introduced graph neural network (Kipf and Welling, 2017) (GNN) into RE model for encoding featrues from dependency tree and proposed a pruning strategy to remove unnecessary components of dependency tree. Guo et al. (2019) also proposed a model with a soft-pruning approach that can automatically learn how to selectively attend to the relevant sub-structures useful for relation extraction.

5.2 RE with Pretrained Models

With the development of pre-trained language models (Devlin et al., 2019), the performance of relation extraction has been highly improved. After that, many researches based on BERT were carried out. Most of these works employ pre-trained language models in three ways for relation extraction: 1) design task-related tasks in pre-training stage to improve prior pattern (Zhang et al., 2019; Joshi et al., 2020; Baldini Soares et al., 2019; Li and Tian, 2020; Peng et al., 2020; Yamada et al., 2020); 2) introduce external knowledge (e.g. knowledge graph and wiki data) into fine-tuning or pre-training stages (Peters et al., 2019a; Baldini Soares et al., 2019; Wang et al., 2020b,a; Yamada et al., 2020); 3) employ representation from pre-trained language models and stack some neural structure over it (Tao et al., 2019; Alt et al., 2019; Wang et al., 2019; Wu and He, 2019b; Shi and Lin, 2019; Zhao et al., 2019; Xue et al., 2020; Chen et al., 2020). There are also some special methods with pre-trained language models (Li et al., 2019; Zhao et al., 2020). They convert relation classification tasks into machine reading comprehension tasks. However, most of them is time-consuming or resource-consuming due to the require of external knowledge and the pre-train stage.

6 Conclusion and Limitations

In this paper, we analyze previous typical works and empirically focus on three granularity features: entity mention level, segment level and sentence level. Based on the hierarchical structure between entity mention level and segment level feature, we propose a multi-granularity hierarchical feature extractor for relation extraction, which does not need any external knowledge or tools. We evaluate our method with different encoders and results on three public benchmarks show that our method can bring outstanding improvement for them.

The structure of our model make it not easy to apply on multi-relation extraction tasks. In the future, we will focus on how to extend our method to longer input tasks and multi-relation extraction tasks (e.g. Document Level Relation Extraction). Besides, we will also investigate what makes graph structure effective in relation extraction tasks and why our method can obtain better results than them.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant Nos.U1636211, 61672081,61370126), the 2020 Tencent Wechat Rhino-Bird Focused Research Program, and the Fund of the State Key Laboratory of Software Development Environment (Grant No. SKLSDE-2021ZX-18).

References

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1558–1569. Association for Computational Linguistics.

Christoph Alt, Marc Hübner, and Leonhard Hennig. 2019. Improving relation extraction by pre-trained language representations. In *Proceedings of AKBC 2019*.

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

Rui Cai, Xiaodong Zhang, and Houfeng Wang. 2016. Bidirectional recurrent convolutional neural network for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 756–765, Berlin, Germany. Association for Computational Linguistics.

Jun Chen, Robert Hoehndorf, Mohamed Elhoseiny, and Xiangliang Zhang. 2020. Efficient long-distance relation extraction with dg-spanbert.

Md. Faisal Mahbub Chowdhury and Alberto Lavelli. 2012. Combining tree structures, flat features and patterns for biomedical relation extraction. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 420–429, Avignon, France. Association for Computational Linguistics.

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

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

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

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

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1148–1158, Portland, Oregon, USA. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Cheng Li and Ye Tian. 2020. Downstream model design of pre-trained language model for relation extraction task.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1340–1350, Florence, Italy. Association for Computational Linguistics.

Jie Liu, Shaowei Chen, Bingquan Wang, Jiaxin Zhang, Na Li, and Tong Xu. 2020. Attention as relation: Learning supervised multi-head self-attention for relation extraction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3787–3793. ijcai.org.

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

Tapas Nayak and Hwee Tou Ng. 2019. Effective attention modeling for neural relation extraction. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 603–612, Hong Kong, China. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48, Denver, Colorado. Association for Computational Linguistics.

Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2020. Learning from Context or Names? An Empirical Study on Neural Relation Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3661–3672, Online. Association for Computational Linguistics.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph lstms. *Trans. Assoc. Comput. Linguistics*, 5:101–115.

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

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019a. 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)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019b. 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)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. N-ary relation extraction using graph-state LSTM. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2226–2235, Brussels, Belgium. Association for Computational Linguistics.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. 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. Association for Computational Linguistics.

Q. Tao, X. Luo, H. Wang, and R. Xu. 2019. Enhancing relation extraction using syntactic indicators and sentential contexts. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1574–1580.

Patrick Verga, Emma Strubell, and Andrew McCallum. 2018. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. 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 872–884, New Orleans, Louisiana. Association for Computational Linguistics.

Ngoc Thang Vu, Heike Adel, Pankaj Gupta, and Hinrich Schütze. 2016. Combining recurrent and convolutional neural networks for relation classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 534–539, San Diego, California. Association for Computational Linguistics.

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

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

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2020a. K-adapter: Infusing knowledge into pre-trained models with adapters.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2020b. Kepler: A unified model for knowledge embedding and pre-trained language representation.

Shanchan Wu and Yifan He. 2019a. Enriching pre-trained language model with entity information for relation classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 2361–2364, New York, NY, USA. Association for Computing Machinery.

Shanchan Wu and Yifan He. 2019b. Enriching pre-trained language model with entity information for relation classification.

Minguang Xiao and Cong Liu. 2016. Semantic relation classification via hierarchical recurrent neural network with attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1254–1263, Osaka, Japan. The COLING 2016 Organizing Committee.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1785–1794, Lisbon, Portugal. Association for Computational Linguistics.

Fuzhao Xue, Aixin Sun, Hao Zhang, and Eng Siong Chng. 2020. Gdpnet: Refining latent multi-view graph for relation extraction.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

Bowen Yu, Zhenyu Zhang, Tingwen Liu, Bin Wang, Sujian Li, and Quangang Li. 2019. 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.

Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2017. 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)*, pages 571–581, Vancouver, Canada. Association for Computational Linguistics.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

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

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

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

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

Tianyang Zhao, Zhao Yan, Yunbo Cao, and Zhoujun Li. 2020. Asking effective and diverse questions: A machine reading comprehension based framework for joint entity-relation extraction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3948–3954. International Joint Conferences on Artificial Intelligence Organization. Main track.

Yi Zhao, Huaiyu Wan, Jianwei Gao, and Youfang Lin. 2019. Improving relation classification by entity pair graph. In *Proceedings of The Eleventh Asian Conference on Machine Learning*, volume 101 of *Proceedings of Machine Learning Research*, pages 1156–1171, Nagoya, Japan. PMLR.

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