Modeling Multi-Granularity Hierarchical Features for Relation Extraction

Anonymous ACL submission

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 even without 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.

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 pre-training 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 entity-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 pre-training 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 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 struc-

\(^1\)entity mentions contain the entity itself and co-references of it.

\(^2\)continuous words in sentence (n-gram)
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 \( x = \{x_1, x_2, ..., x_n\} \) be a sequence of input tokens, where \( x_0 = \text{CLS} \) and \( x_n = \text{SEP} \) 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 \( x: [x_i, ..., x_{j-1}] \) and \( [x_k, ..., x_{l-1}] \). Our goal is to learn a function \( P(r) = f_\theta(x, s_1, s_2) \), where \( r \in \mathcal{R} \) indicates the relation between the entity pairs, which is marked by \( s_1 \) and \( s_2 \). \( \mathcal{R} \) 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).

\[
H = \{h_0, ..., h_n\} = f_{\text{encoder}}(x_0, ..., x_n) \tag{1}
\]

Where \( H = \{h_0, ..., h_n\} \) is the vector representation of input sentences.

Our work is built upon \( H \) and does not need any external information. We employ a max-pooling operation to obtain shallow features of entity pairs and input sentences. \( h_{e_1} = \text{Maxpooling}(h_{e_1}) \) and \( h_{e_2} = \text{Maxpooling}(h_{e_2}) \) are the representations of entity pairs. \( h_y = \text{Maxpooling}(H) \) 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 $H$ by Equ. (2).

$$h'_{e_1} = \text{Softmax}(\frac{H \cdot h_{e_1}}{\sqrt{d}}) \cdot H$$

$$h'_{e_2} = \text{Softmax}(\frac{H \cdot h_{e_2}}{\sqrt{d}}) \cdot H$$  \hspace{1cm} (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).

$$H_t = \text{CNN}_t(H), t \in \{1, 2, 3\}$$  \hspace{1cm} (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 $H_t$ 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).

$$h'_m = \text{Softmax} \left( \frac{H_t \cdot (W_m[h'_1; h'_2])}{\sqrt{d}} \right) \cdot H_t \quad (4)$$

Then, we get $\{h'_m\}_{t=1,2,3}$ 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 $[h_{e1}; h_{e2}; h_g]$ 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 $[h_{e1}; h_{e2}; h_g]$ as query to capture deeper semantic feature from context representation $H$ by Equ. (5).

$$h_s = \text{Softmax} \left( \frac{H \cdot (W_s[h_{e1}; h_{e2}; h_g])}{\sqrt{d}} \right) \cdot H \quad (5)$$

Where $W_s \in \mathbb{R}^{d \times 3d}$ is a linear transform matrix, and $d$ is a hidden dimension of vectors. The concatenation of $[h_{e1}; h_{e2}; h_g]$ 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).

$$h_o = \text{ReLU}(W_o[h_s; h'_m; h'_1; h'_m; h''_1; h''_m]) \quad (6)$$

Where $W_o \in \mathbb{R}^{6d \times d}$ 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:

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

The whole model is trained with cross entropy loss function. We call the multi-granularity hierarchical feature extractor: SMS (relation extraction with Sentence level, Mention level and mention-aware Segment level features).

### 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** 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** 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-

<table>
<thead>
<tr>
<th>Hyper-parameters</th>
<th>Tacred</th>
<th>Semeval</th>
</tr>
</thead>
<tbody>
<tr>
<td>lr</td>
<td>3e-5</td>
<td>2e-5</td>
</tr>
<tr>
<td>warmup steps</td>
<td>300</td>
<td>2</td>
</tr>
<tr>
<td>batch size</td>
<td>64</td>
<td>32</td>
</tr>
<tr>
<td>V100 GPU</td>
<td>4x</td>
<td>1x</td>
</tr>
<tr>
<td>epochs</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>max length</td>
<td>128</td>
<td>128</td>
</tr>
</tbody>
</table>

3https://nlp.stanford.edu/projects/tacred/  
4https://github.com/DFKI-NLP/tacrev
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

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 knowledge graphs.

<table>
<thead>
<tr>
<th>Model</th>
<th>Tacred</th>
<th>Tacred Revisited</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P(ΔP)</td>
<td>R(ΔR)</td>
</tr>
<tr>
<td>LSTM</td>
<td>62.5</td>
<td>63.4</td>
</tr>
<tr>
<td>PA-LSTM*</td>
<td>65.7</td>
<td>64.5</td>
</tr>
<tr>
<td>1 SA-LSTM</td>
<td>68.1</td>
<td>65.7</td>
</tr>
<tr>
<td>C-GCN*</td>
<td>69.9</td>
<td>63.3</td>
</tr>
<tr>
<td>C-AGGCN*</td>
<td><strong>71.9</strong></td>
<td>63.4</td>
</tr>
<tr>
<td>TRE</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BERT-base</td>
<td>68.1</td>
<td>67.7</td>
</tr>
<tr>
<td>BERT-large</td>
<td>69.2</td>
<td>69.4</td>
</tr>
<tr>
<td>2 BERT+LSTM</td>
<td>73.3</td>
<td>63.1</td>
</tr>
<tr>
<td>SpanBERT-base</td>
<td>67.6</td>
<td>68.6</td>
</tr>
<tr>
<td>SpanBERT-large</td>
<td>70.8</td>
<td>70.9</td>
</tr>
<tr>
<td>DG-SpanBERT-large*</td>
<td><strong>71.4</strong></td>
<td><strong>71.6</strong></td>
</tr>
<tr>
<td>MTB†</td>
<td>71.6</td>
<td>71.4</td>
</tr>
<tr>
<td>KnowBERT-W+W‡</td>
<td>71.0</td>
<td>74.0</td>
</tr>
<tr>
<td>3 KEPLER†</td>
<td>70.4</td>
<td>73.0</td>
</tr>
<tr>
<td>K-Adapter‡</td>
<td>70.1</td>
<td>74.0</td>
</tr>
<tr>
<td>LUKE†</td>
<td>70.4</td>
<td>75.1</td>
</tr>
</tbody>
</table>

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

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 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.

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.
Table 4: Ablation study on Tacred Revisited test set.

| Feature Level | F1  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SpanBERT-large</td>
<td>78.0</td>
</tr>
<tr>
<td>+ sentence level</td>
<td>78.6 (+0.6)</td>
</tr>
<tr>
<td>+ mention level</td>
<td>78.8 (+0.8)</td>
</tr>
<tr>
<td>+ segment level</td>
<td>79.4 (+1.4)</td>
</tr>
<tr>
<td>+ all</td>
<td>79.8 (+1.8)</td>
</tr>
</tbody>
</table>

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.

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
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 features 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., 2020a; 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.
References


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


603–612, Hong Kong, China. Association for Computational Linguistics.


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


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.


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


