Document-Level Relation Extraction via Pair-Aware and Entity-Enhanced Representation Learning

Xiusheng Huang\textsuperscript{1,2}, Hang Yang\textsuperscript{1,2}, Yubo Chen\textsuperscript{1,2}, Jun Zhao\textsuperscript{1,2}, Kang Liu\textsuperscript{1,2,4}, Weijian Sun\textsuperscript{3} and Zuyu Zhao\textsuperscript{3}

\textsuperscript{1}School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China  
\textsuperscript{2}National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing, China  
\textsuperscript{3}Huawei Technologies Co., Ltd, Shenzhen, China  
\textsuperscript{4}Beijing Academy of Artificial Intelligence, Beijing, China  

huangxiusheng2020@ia.ac.cn, \{hang.yang,yubo.chen,jzhao,kliu\}@nlpr.ia.ac.cn, \{sunweijian,zhaozuyu1\}@huawei.com

Abstract

Document-level relation extraction aims to recognize relations among multiple entity pairs from a whole piece of article. Recent methods achieve considerable performance but still suffer from two challenges: a) the relational entity pairs are sparse, b) the representation of entity pairs is insufficient. In this paper, we propose Pair-Aware and Entity-Enhanced (PAEE) model to solve the aforementioned two challenges. For the first challenge, we design a Pair-Aware Representation module to predict potential relational entity pairs, which constrains the relation extraction to the predicted entity pairs subset rather than all pairs; For the second, we introduce a Entity-Enhanced Representation module to assemble directional entity pairs and obtain a holistic understanding of the entire document. Experimental results show that our approach can obtain state-of-the-art performance on four benchmark datasets DocRED, DWIE, CDR and GDA.

1 Introduction

Relation extraction (RE) is a primary task in the field of information extraction, which aims to identify the relationships between two entities in a document. Previous works mainly focus on sentence-level relation extraction, i.e, recognizing the relationships between entities in a sentence. However, large amounts of relationships are expressed over multiple sentences in real-world applications. According to DocRED (Yao et al., 2019), above 40.7% of the relational facts can only be extracted from multiple sentences. Therefore, it requires the model to capture complex interactions among entities in the whole document. Previous work commonly referred to this problem as document-level relation extraction which has attracted much attention recently (Nan et al., 2020; Zhou et al., 2021; Zhang et al., 2021). Although the considerable performance of these methods, there are still two critical challenges in document-level RE to be addressed.

![Fig. 1: An example with entity pairs and relations from DocRED. Entity mentions only involved in these relation instances are colored, other entities in the document are highlighted in grey.](image)

The first challenge is how to identify relational entity pairs that are sparse in a document. Specifically, given a document with \( n \) entities, there will be \( n(n - 1) \) combinations of entities to classify. However, only a few entity pairs have predefined relationships. For example, as shown in Figure 1, this document contains 21 entities with 420 potential entity pairs. However, the number of relational entity pairs is only 11, accounting for 2.62% of the total entity pairs. According to statistics, for DocRED (Yao et al., 2019) dataset, the proportion are 3.18% and 3.11% in the train set and dev set, respectively. To further explore the impact of sparsity on performance bottlenecks, we conduct a diagnostic experiment on DocRED dataset. Utilizing previous SOTA model ATLOP (Zhou et al., 2021), we divulge the information of whether existing a predefined relationship between the entities to classify. Specifically, we just concatenate a 0-1 variable on the original representation of entity pairs, where “1” represents the entity pair exists a predefined relationship.
relationship. Experimental results show that the F1 score reaches 93.50% in dev set which is 32.20% higher than normal setting. This demonstrates that the importance of identifying the relational entity pairs when facing the sparsity problem (Wang et al., 2019a).

The second challenge is how to effectively model the representations of entity pairs. There are commonly two characteristics for entity pairs. Firstly, the entities-scattering, which means the entities of an entity pair may scatter across multiple sentences. Figure 1 illustrates an example from the DocRED dataset. For Pair_A, the subject Ali Abdullah Ahmed and object Yasir al-Salami are distributed in different sentences ([S1] and [S7]), which requires model to capture the long-distance dependency among entities across sentences. Secondly, the directivity of entity pairs, which means that the relationships of entity pairs are directional. For example, the Pair_B and Pair_B′ in the Figure 1, their subject and object are opposite, and the relations of them are different. Therefore, this challenge requires the model to assemble directional entity pairs and obtain a holistic understanding of the cross-sentence context. To model the representation of entity pairs, most current approaches include graph-based methods and transformer-based methods. Specifically, some methods (Christopoulou et al., 2019; Nan et al., 2020; Wang et al., 2020) construct a document graph with structured attention, dependency structures or heuristics. Meanwhile, considering the transformer can capture long-distance information, some studies (Wang et al., 2019a; Tang et al., 2020; Zhou et al., 2021) directly apply pre-trained language models without introducing graph structures. However, they directly concatenate two entities together to obtain the representation of entity pair, without considering the directivity of entity pairs and modeling the representations of entity pairs adequately.

In this paper, we propose a Pair-Aware and Entity-Enhanced (PAEE) model for document-level RE. To deal with the sparsity of entity pairs, we propose the Pair-Aware Representation (PAR) module to identify potential relational entity pairs, which constrains the relation extraction to the predicted pairs subset rather than all pairs. Furthermore, to capture the global features of triples, PAR utilizes TNet (Papadopoulos et al., 2021) to model the relation between entity pairs, unlike previous methods, PAR designs a Sliding Window Filling Strategy for filling relation matrix, which enhances the interaction between entity pairs. To effectively model the representation of entity pairs, we focus on the global interactions among sentences and entities. Specifically, we propose a Entity-Enhanced Representation (EER) module. The EER first introduces a Representation-Enhanced Encoder to facilitate the interaction between all sentences and entities. In this way, EER obtains a holistic understanding of the entire document. Then, considering that the characteristics of entities as subjects and objects are different, especially in different relationship categories, EER utilizes a Cross Matching method to assemble directional entity pairs.

Experiments on four document-level relation extraction datasets, DocRED (Yao et al., 2019), DWIE (Zaporojets et al., 2021), CDR (Li et al., 2016) and GDA (Wu et al., 2019), demonstrate that our PAEE model significantly outperforms the state-of-the-art methods. To our best knowledge, we are first to consider the sparsity and the directivity of relational entity pairs for the task.

We summarize our contributions as follows:

- To alleviate the negative impact of sparsity, we propose Pair-Aware Representation (PAR) module, which promotes the interaction between entity pairs and accurately identifies potential relational entity pairs.
- To model the representation of entity pairs better, we propose Entity-Enhanced Representation (EER) module, which is based on a Representation-Enhanced Encoder to capture the global context for the scattered entities and a Cross Matching method to assemble directional entity pairs.
- We conduct experiments on four public document-level relation extraction datasets. Experimental results demonstrate that our PAEE model can achieves state-of-the-art performance compared with baselines.

2 Methodology

Before introducing our proposed approach for PAEE in this section, we first introduce the problem definition. Given a document \(d\) and a set of entities \(\{e_i\}_{i=1}^n\), and there are \(n(n - 1)\) entity
pairs in this document. The task of document-level relation extraction is to predict a subset of relations from \( R \cup \{\text{NA}\} \) between the entity pairs \((e_s, e_o)\), where \( R \) is a pre-defined set of relationships, \( e_s, e_o \) are identified as subject and object entities, respectively. The entity pairs that do not express any relation are labeled NA. In addition, the model needs to predict the label of all entity pairs \((e_s, e_o)\) at the test time. To model relation extraction between \( e_s \) and \( e_o \), we define a \( N \times N \) matrix \( V \), where entry \( V_{s,o} \) indicates the relation type between \( e_s \) and \( e_o \). Entities in \( V \) are arranged according to their first appearance in the document. Unlike Zhang et al. (2021), we utilize the sliding window filling strategy to fill matrix \( V \), which can enhance the interaction between entity pairs and is beneficial to relation extraction.

2.1 Encoder

Given a document \( D = \{x_i\}_{i=1}^l \) with \( l \) tokens, we insert special symbols “,” “<”, “>” and “?” to mark the entity positions at the start and end of mentions. It is adapted from the entity marker technique (Zhang et al., 2017; Shi and Lin, 2019; Soares et al., 2019). We leverage the pre-trained language model as an encoder to obtain the embedding as follows:

\[
H = [h_1, h_2, ..., h_l] = \text{Encoder}([x_1, x_2, ..., x_l])
\]

(1)

where \( h_i \) is the embedding of the token \( x_i \). Note that some documents are longer than 512, we thus leverage a dynamic window to encode whole documents (Zhou et al., 2021). We take the embedding of “<” “>” at the start of mentions as the mention embeddings. Then, for an entity \( e_i \) with mentions \( \{m_j^i\}_{j=1}^{N_i} \), we leverage a logsumexp pooling to obtain the entity embedding \( e_i \):

\[
e_i = \log \sum_{j=1}^{N_i} \exp(m_j)
\]

(2)

This pooling accumulates signals from mentions in the document. Compared with the mean pooling, the logsumexp pooling shows better performance in the experiment. We calculate the entity-level relation matrix based on entity-to-entity relevance. Specifically, we constructed a \( D \)-dimensional feature vector \( V(e_s, e_o) \) to capture the relevance between entities. Note that we add the position and type information of entities to enrich the vector \( V(e_s, e_o) \). For intra-sentential and inter-sentential entity pairs, their position captured by a 0-1 variable \( pos \).
where $W_1$ is the learnable weight matrix, $a_{(s,o)}$ is the attention weight of last layer for entity-aware attention and $A_i^s$ refers to the tokens’ importance to the $i$-th entity, $H$ is the contextual embedding in Eq.1. The $K$ is the number of head in the transformer.

### 2.2 Pair-Aware Representation

In this section, we propose Pair-Aware Representation (PAR) module to enhance the interaction between entity pairs and identify potential entity pairs. We build the module base on existing BERT baselines (Zhou et al., 2021; Zhang et al., 2021) and integrate other techniques to further improve the performance.

**Sliding Window Filling Strategy.** To capture the relevance of entity pairs, we utilize TNet (Papadopoulos et al., 2021) to expand receptive field and learn more global and local information. The TNet is a novel multi-scale hard-attention architecture that constantly adjusts the number of elements to help us focus on the related entity pairs. We take the matrix $V \in \mathbb{R}^{N \times N \times D}$ as a $D$-channel variable and feed it into TNet, where $N$ is the largest number of entities, counted from all the dataset samples. However, the number of entities annotated in each document is usually different and often less than $N$, thus, we propose a sliding window filling strategy to fill matrix before feeding matrix $V$ into TNet.

\[
V' = \text{Sliding}(V), \\
Y = \text{TNet}(W_2 V')
\]  

where $Y \in \mathbb{R}^{N \times N \times D'}$ denotes the entity-level relation matrix. $W_2$ is the learnable weight matrix and $D'$ is much smaller than $D$. As it shows in the Figure 2, the diagonal dots are far apart in the matrix $V$, which makes their interaction poor (Ronneberger et al., 2015). Instead of previous zero filling (Ronneberger et al., 2015), we utilize the sliding window filling strategy to shorten the distance between entity pairs. Specifically, for the orange dashed window in matrix $V$, we slide the window in three directions: transverse, longitudinal and oblique, then we will obtain a filled matrix $V'$. Furthermore, in the whole matrix $V'$, the spacing between dots that were originally far away was significantly shortened, which facilitates the interaction between them.

**Potential Pair Prediction.** This component is shown as a 0-1 distribution box in Figure 2, where “1” means potential relational entity pairs. Given a document which contains multiple entity pairs, different from previous works (Zhou et al., 2021; Zhang et al., 2021) which redundantly perform relationship classification to every entity pair, we utilize this module to predict potential relational entity pairs. Specifically, we utilize the average pooling operation (Lin et al., 2013) to obtain the representation $\mathbf{P}_{pair}$ of each entity pair, and then feed it into the binary classifier to get the potential entity pairs.

\[
\mathbf{P}_{pair}^i = \kappa(\mathbf{P}_{pair}; \lambda; \text{pos}; \text{sub}_\text{emb}; \text{obj}_\text{emb})
\]  

where $\kappa$ and $\lambda$ denote the binary classifier and threshold, $\text{sub}_\text{emb}$ is the type embedding of subject in entity pairs, $\text{obj}_\text{emb}$ is the type embedding of object in entity pairs. We model it as a binary classification task, and the corresponding entity pairs will be assigned with tag “1” if the probability exceeds a certain threshold $\lambda$ or with tag “0” otherwise (as shown in Figure 2). By concatenating the classification results $\mathbf{P}_{pair}$ with matrix $Y$ in Eq.4, we will obtain a entity-level relation matrix $Y_{pair} \in \mathbb{R}^{D'+1}$ incorporating the information of candidate pairs.

### 2.3 Entity-Enhanced Representation

In this section, we propose a Entity-Enhanced Representation module to model the representation of entity pairs. Specifically, we introduce Representation-Enhanced Encoder to facilitate the interaction between all sentences and entities. Then, considering that the characteristics of entity as subject and object are different, especially in different relational categories, we propose Cross Matching method to assemble directional entity pairs.

**Representation-Enhanced Encoder.** To enable the awareness of document-level contexts for sentences and entities, we employ a Representation-Enhanced Encoder to facilitate the interaction between all sentences and entities. Formally, we can obtain the entity embedding $\mathbf{e}_i$ from Eq.2 and
the embedding $[h_1, h_2, \ldots, h_l]$ of every token in sentence $S_i$ from Eq.1, where $l$ is the sentence length. Hence the sentence embedding $S_i$ can be obtained by a max-pooling operation over the token sequence representation. Then we employ the Transformer (Vaswani et al., 2017) module, Representation-Enhanced Encoder, as the encoder to obtain the document-aware embedding for sentences and entities. Note that we add the sentence representation with sentence position embeddings to inform the sentence order before feeding them into the Representation-Enhanced Encoder.

$$[H^e; H^o] = \text{RE-Encoder}(e_1 \ldots e_{N_e}; S_1 \ldots S_{N_s})$$

where $S_i$ is the local representation for $i$-th sentence and $e_i$ is the representation for $i$-th entity. Utilizing the Representation-Enhanced Encoder, we can obtain the document-aware entities representation $H^e \in \mathbb{R}^{N_e \times D}$, $N_e$ is the number of entities in a document.

**Cross Matching.** To extract the different features of entity as subject and object respectively, we utilize the Sub-Obj layer (a Linear Layer$(N_e \times D, 2 \times N_e \times N_c)$) for feature separation. Meanwhile, we map these features to each relationship category, which enhances the interaction between entities in each relationship. For the Sub-Obj layer, we set a corresponding loss (Appendix A.1) to learn that a single entity may have several relationships. The features of entity as subject and object in each relationship can be calculated as:

$$[F_{sub}; F_{obj}] = \text{Sub-Obj}(H^e)$$

where $F_{sub}, F_{obj} \in \mathbb{R}^{N_c \times N_e}$ denotes the features of entities as subjects and objects respectively, $N_c$ is the number of relationship categories and $N_e$ is the number of entities. Meanwhile, we concatenate these features with the representation $H^e$ of entities, then we will obtain $e_{sub}, e_{obj} \in \mathbb{R}^{N_e \times (D+N_c)}$, which are the representations of entities as subjects and objects respectively.

**Classification Module.** Given the entity embedding $e_{sub}$ and $e_{obj}$ with entity-level relation matrix $Y_{pair}$ in section 2.2, we map them to hidden representations $z$ with a feedforward neural network. Then we calculate the probability of relation $r$ by bilinear function and sigmoid activation. Formally, we obtain:

$$z_s = \tanh(W_s \cdot e_{sub} + Y_{s,o}),$$

$$z_o = \tanh(W_o \cdot e_{obj} + Y_{s,o}),$$

$$P(r \mid e_{sub}, e_{obj}) = \sigma(z_s W_r z_o + b_r)$$

where $Y_{s,o}$ is the entity-pair representation of $(s, o)$ in matrix $Y_{pair}$, $\sigma$ denotes the sigmoid function, $W_s \in \mathbb{R}^{d \times d}$, $W_o \in \mathbb{R}^{d \times d}$, $b \in \mathbb{R}$, and $W_r \in \mathbb{R}^{d\times d}$ are learnable parameters.

### 3 Experiments

#### 3.1 Experimental Setup

**Datasets.** We evaluated our method on four document-level RE datasets. The statistical results of the datasets are shown in Table 2.

- **DocRED** (Yao et al., 2019) is a large-scale document-level relation extraction dataset. It is constructed from Wikipedia articles. DocRED contains 96 relationships and 3,053/1,000/1,000 instances for training, validating and test, respectively.
- **DWIE** (Zaporojets et al., 2021) is a document-level RE dataset after processing. This dataset has 700 documents for train and 99 documents for test. The training set is then randomly split into two parts: 602 documents for train and 98 for development.
- **CDR** (Li et al., 2016) is a relation extraction dataset in the biomedical domain, which is human-annotated and aims to predict the binary interactions between Chemical and Disease concepts.
- **GDA** (Wu et al., 2019) is a large-scale dataset in the biomedical domain, which aims to predict the binary interactions between Gene and Disease concepts.

**Pretrained Transformers.** We initialize PAEE with three different pretrained language models including BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and SciBERT (Beltagy et al., 2019).

<table>
<thead>
<tr>
<th>Statistics/Datasets</th>
<th>DocRED</th>
<th>DWIE</th>
<th>CDR</th>
<th>GDA</th>
</tr>
</thead>
<tbody>
<tr>
<td># Train</td>
<td>3,053</td>
<td>602</td>
<td>500</td>
<td>23,353</td>
</tr>
<tr>
<td># Dev</td>
<td>1,000</td>
<td>98</td>
<td>500</td>
<td>5,839</td>
</tr>
<tr>
<td># Test</td>
<td>1,000</td>
<td>99</td>
<td>500</td>
<td>1,000</td>
</tr>
<tr>
<td># Relation</td>
<td>97</td>
<td>65</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Avg. # entity per Doc</td>
<td>19.5</td>
<td>14</td>
<td>7.6</td>
<td>5.4</td>
</tr>
<tr>
<td>Avg. # Ment. per Ent.</td>
<td>1.4</td>
<td>1.6</td>
<td>2.7</td>
<td>3.3</td>
</tr>
</tbody>
</table>

Table 1: Statistics of the experimental datasets.
Table 2: Main results on the development and test set of DocRED. We report the official test score on the CodaLab scoreboard with the best checkpoint on the development set. The performance of our method is followed by the improvements (↑) over the previous state-of-the-art method DocuNet.

<table>
<thead>
<tr>
<th>Model</th>
<th>Dev (%)</th>
<th>Test (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ig F1 (%)</td>
<td>F1 (%)</td>
</tr>
<tr>
<td>BERT\textsubscript{base} (Wang et al., 2019b)</td>
<td>-</td>
<td>54.16</td>
</tr>
<tr>
<td>BERT-\textsubscript{T5}base (Wang et al., 2019a)</td>
<td>-</td>
<td>54.42</td>
</tr>
<tr>
<td>HIN-BERT\textsubscript{base} (Tang et al., 2020)</td>
<td>54.29</td>
<td>56.31</td>
</tr>
<tr>
<td>CorefBERT\textsubscript{base} (Ye et al., 2020)</td>
<td>55.32</td>
<td>57.51</td>
</tr>
<tr>
<td>SSAN-BERT\textsubscript{base} (Xu et al., 2021a)</td>
<td>57.03</td>
<td>59.19</td>
</tr>
<tr>
<td>ATLOP-BERT\textsubscript{base} (Zhou et al., 2021)</td>
<td>59.22</td>
<td>61.09</td>
</tr>
<tr>
<td>DocuNet-BERT\textsubscript{base} (Zhang et al., 2021)</td>
<td>59.86</td>
<td>61.28</td>
</tr>
<tr>
<td>PAEE-BERT\textsubscript{base} (Ours)</td>
<td>60.38↑ (70.52)</td>
<td>62.62↑ (71.34)</td>
</tr>
<tr>
<td>BERT\textsubscript{large} (Ye et al., 2020)</td>
<td>56.67</td>
<td>58.83</td>
</tr>
<tr>
<td>CorefBERT\textsubscript{large} (Ye et al., 2020)</td>
<td>58.62</td>
<td>59.01</td>
</tr>
<tr>
<td>RoBERTa\textsubscript{large} (Ye et al., 2020)</td>
<td>57.14</td>
<td>59.22</td>
</tr>
<tr>
<td>CorefRoBERTa\textsubscript{large} (Ye et al., 2020)</td>
<td>57.35</td>
<td>59.43</td>
</tr>
<tr>
<td>SSAN-RoBERTa\textsubscript{large} (Xu et al., 2021a)</td>
<td>60.25</td>
<td>62.08</td>
</tr>
<tr>
<td>ATLOP-RoBERTa\textsubscript{large} (Zhou et al., 2021)</td>
<td>61.32</td>
<td>63.18</td>
</tr>
<tr>
<td>DocuNet-RoBERTa\textsubscript{large} (Zhang et al., 2021)</td>
<td>61.43</td>
<td>63.40</td>
</tr>
<tr>
<td>PAEE-RoBERTa\textsubscript{large} (Ours)</td>
<td>62.44↑ (71.01)</td>
<td>64.82↑ (71.42)</td>
</tr>
</tbody>
</table>

- **BERT** employs a Transformer encoder to learn from large unlabeled text corpora and sub-word units to represent textual tokens, which contains 12 and 24 self-attention layers.

- **RoBERTa** is an improved version of BERT, which removes the Next Sentence Prediction task and adopts larger text corpora as well as more training steps.

- **SciBERT** adopts the same model architecture as BERT, but is trained on scientific text instead. In this paper, we provide SciBERT-initialized PAEE on the two biomedical domain datasets CDR and GDA.

**Implementation Detail.** We used cased BERT-base, or RoBERTa-large as the encoder on DocRED and SciBERT-base on CDR and GDA. We use mixed-precision training (Micikevicius et al., 2018) based on the Apex library. Our model is optimized with AdamW (Loshchilov and Hutter, 2018) using learning rates $\epsilon \in \{2e^{-5}, 3e^{-5}, 5e^{-5}, 1e^{-4}\}$, with a linear warmup (Goyal et al., 2018) for the first 6 steps followed by a linear decay to 0. We set the matrix size $N=42$ in the Figure 2 and $\lambda = 0.3$. We preprocess CDR and GDA dataset following Christopoulou et al. (2019). For GDA, we split 20% of the training set for development. For CDR, we merge the training set and dev set to train the final model after the best hyper-parameter is set. The calculation of loss will be provided in the appendix A.1. We report the mean and standard deviation of F1 on the development set by conducting 5 runs of training using different random seeds.

**Evaluation.** Our primary evaluation metric are F1, Ign F1 (Yao et al., 2019) and Rela. Ign F1 is computed by excluding relational facts that already appeared in the training set. It avoids information leakage from the training set. We propose Rela for evaluating the accuracy of identifying relational entity pairs. The prediction results of entity pairs are processed into two classification tasks. The relationship between entity pairs is divided into NA and non NA.

**3.2 Experiment Results**

We conduct experiments on four DocRE datasets to verify the effectiveness of our method PAEE.

**Results on the DocRED Dataset.** In the DocRED dataset, we compare PAEE with transformer-based models, including BERT\textsubscript{base} (Wang et al., 2019b), BERT-\textsubscript{T5}base (Wang et al., 2019a), CorefBERT\textsubscript{base} (Ye et al., 2020), HIN-BERT\textsubscript{base} (Tang et al., 2020), SSAN (Xu et al., 2021a) and ATLOP\textsubscript{base} on the DocRED dataset; and graph-based models, including GEDA (Li et al., 2020), LSR (Nan et al., 2020), GLRE (Wang et al., 2020), GAIN (Zeng et al., 2020), HeterGSAN (Xu et al., 2021b) and DocuNet (Zhang et al., 2021). Results in Table 2 shows that PAEE performs better than these methods. Our best model, PAEE built upon RoBERTa\textsubscript{large}, is $+1.42 / +1.57$ F1 better on dev/test set than DocuNet-RoBERTa\textsubscript{base}. 
Table 3: Main results on the development and test set of DWIE. The performance of our method is followed by the improvements (↑) over the previous state-of-the-art method ATLOP.

<table>
<thead>
<tr>
<th>Model</th>
<th>Dev Ign F1 (%)</th>
<th>Dev F1 (%)</th>
<th>Dev Rela (%)</th>
<th>Test Ign (%)</th>
<th>Test F1 (%)</th>
<th>Test Rela (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>37.65</td>
<td>47.73</td>
<td>56.43</td>
<td>34.65</td>
<td>46.14</td>
<td>55.83</td>
</tr>
<tr>
<td>LSTM</td>
<td>40.86</td>
<td>51.77</td>
<td>59.31</td>
<td>40.81</td>
<td>52.60</td>
<td>61.42</td>
</tr>
<tr>
<td>BiLSTM</td>
<td>40.46</td>
<td>51.92</td>
<td>59.49</td>
<td>42.03</td>
<td>54.47</td>
<td>64.78</td>
</tr>
<tr>
<td>GAIN</td>
<td>58.63</td>
<td>62.55</td>
<td>74.75</td>
<td>62.37</td>
<td>67.57</td>
<td>78.89</td>
</tr>
<tr>
<td>ATLOP</td>
<td>59.03</td>
<td>64.82</td>
<td>77.43</td>
<td>62.09</td>
<td>69.94</td>
<td>82.12</td>
</tr>
<tr>
<td>PAEE (Ours)</td>
<td>62.05 (↑3.02)</td>
<td>67.52 (↑2.70)</td>
<td>82.01 (↑4.58)</td>
<td>66.45 (↑4.36)</td>
<td>73.10 (↑4.33)</td>
<td>86.45 (↑4.33)</td>
</tr>
</tbody>
</table>

Table 4: Test F1 score (%) on CDR and GDA dataset. Our PAEE model with the SciBERT encoder outperforms the current state-of-the-art results. The performance of our method is followed by the improvements (↑) over the previous state-of-the-art method DocuNet.

<table>
<thead>
<tr>
<th>Model</th>
<th>CDR</th>
<th>GDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRAN</td>
<td>62.1</td>
<td>-</td>
</tr>
<tr>
<td>LSR</td>
<td>64.8</td>
<td>82.2</td>
</tr>
<tr>
<td>DHG</td>
<td>65.9</td>
<td>83.1</td>
</tr>
<tr>
<td>GLRE</td>
<td>68.5</td>
<td>-</td>
</tr>
<tr>
<td>SciBERT (Belty et al., 2019)</td>
<td>65.1</td>
<td>82.5</td>
</tr>
<tr>
<td>SSAN SciBERT (Xu et al., 2021a)</td>
<td>68.7</td>
<td>83.7</td>
</tr>
<tr>
<td>ATLOP SciBERT (Zhou et al., 2021)</td>
<td>69.4</td>
<td>83.9</td>
</tr>
<tr>
<td>DocuNet SciBERT (Zhang et al., 2021)</td>
<td>76.3</td>
<td>85.3</td>
</tr>
<tr>
<td>PAEE-SciBERT</td>
<td>78.2 (↑1.9)</td>
<td>87.3 (↑2.4)</td>
</tr>
</tbody>
</table>

Table 5: Ablation study of PAEE on DocRED. We turn off different components of the model one at a time.

<table>
<thead>
<tr>
<th>Model</th>
<th>Ign F1 (%)</th>
<th>F1 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAEE-BERT base</td>
<td>60.38</td>
<td>62.62</td>
</tr>
<tr>
<td>w/o PAR</td>
<td>57.67 (↓2.71)</td>
<td>59.61 (↓3.01)</td>
</tr>
<tr>
<td>w/o EER</td>
<td>59.57 (↓0.81)</td>
<td>61.53 (↓1.09)</td>
</tr>
<tr>
<td>w/o SW</td>
<td>59.72 (↓0.66)</td>
<td>61.72 (↓0.90)</td>
</tr>
<tr>
<td>w/o PPP</td>
<td>59.63 (↓0.75)</td>
<td>61.52 (↓1.10)</td>
</tr>
</tbody>
</table>

3.3 Ablation Study

To show the efficacy of our proposed techniques, we conduct an ablation study experiment by turning off one component at a time. 1) w/o PAR, which removes the Pair-Aware Representation module; 2) w/o EER, which removes the Entity-Enhanced Representation module, we directly splice two entities as the representation of entity pairs; 3) w/o SW, which removes the Sliding Window strategy, the previous zero filling method is used to fill the whole relationship matrix; 4) w/o PPP, which removes the Potential Pair Prediction module. We present the results of ablation study in Table 5. From the results, we can observe that:

1. **Effectiveness of Pair-Aware Representation.** When we remove the Pair-Aware Representation module from the PAEE, the F1 score drops by 3.01% on DocRED dataset. It proves the Pair-Aware Representation module is very effective for the task.

2. **Effectiveness of Entity-Enhanced Representation.** Compared with the model removed Entity-Enhanced Representation module, our method PAEE achieves 1.09% improvements
Johan Gottlieb Gahn (19 August 1745 – 8 December 1818) was a Swedish chemist who discovered manganese in 1774. Gahn studied in Uppsala 1762–1770 and became acquainted with chemists Torbern Bergman and Carl Wilhelm Scheele. In 1770 he settled in Falun, where he introduced improvements in copper smelting, and participated in building up several factories, including those for vitriol, sulfur and red paint.

Table 6: The ACC means the accuracy of identifying relational entity pairs.

<table>
<thead>
<tr>
<th>Model</th>
<th>ACC</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT_{base}</td>
<td>48.83</td>
<td>54.16</td>
</tr>
<tr>
<td>CorefBERT</td>
<td>59.37</td>
<td>57.51</td>
</tr>
<tr>
<td>ATLOP</td>
<td>65.42</td>
<td>61.09</td>
</tr>
<tr>
<td>PAEE-BERT_{base} (Our)</td>
<td>70.30</td>
<td>64.82</td>
</tr>
</tbody>
</table>

Fig. 3: Case study on our proposed PAEE and baseline model. Entity mentions only involved in these relation instances are colored, other entities in the document are high-lighted in grey. We utilize arrows to connect relational entity pairs.

We observe that for both models, their performance gets better when the proportion of relational entity pairs becomes larger, and the model w/ EER consistently outperforms the model w/o EER. This demonstrates that EER can model the representation of entity pairs better.

3.4 Discussion and Analysis

In order to explore whether the performance bottleneck of the model is effectively solved, we utilize experiments to analyze it.

The effect of PAEE on sparsity. To assess the effectiveness of PAEE on identifying relational entity pairs, we analyze it from contrast experiments, the experiments are based on the pre-training model BERT_{base} and DocRED dataset. As shown in Table 6, the ACC score if 70.3% which is 4.88% more than previous SOTA model ATLOP. This shows that PAEE model can effectively identify potential relational entity pairs.

The effect of Entity-Enhanced Representation (EER). To show that our EER can model the representation of entity pairs better, we divide the documents in dev set of DocRED into different groups by the proportion of relational entity pairs, and evaluate models trained with or without the EER. Experiment results are shown in Figure 4.

3.5 Case Study

We select a sample from the dev set of the DocRED dataset and conduct a case study to further illustrate the effectiveness of our model PAEE compared with the baseline. As shown in Figure 3, we notice that both BERT_{base} and PAEE-BERT_{base} can successfully extract the “Country of citizenship” relation between “Gahn” and “Swedish”. However, only our PAEE-BERT_{base} can deduce that the “Country” of “Uppsala” and “Falun” are same, namely “Swedish”.

4 Conclusion

In this paper, we propose the Pair-Aware and Entity-Enhanced (PAEE) model. Specifically,
PAEE introduces Pair-Aware Representation (PAR) module to alleviate the negative impact of sparsity, which constrains the following relation extraction to the predicted entity pairs subset rather than all pairs. In addition, PAEE also designs Entity-Enhanced Representation (EER) module to assemble directional entity pairs and obtain holistic understanding of document. Experiments on four benchmark datasets DocRED, DWIE, CDA and GDA, show that PAEE outperforms the previous methods and obtains new state-of-the-art results.

5 Acknowledgements

This work is supported by the National Key Research and Development Program of China (No.2020AAA0106400), the National Natural Science Foundation of China (No.61922085, 61976211, 62176257). This work is also supported by the Strategic Priority Research Program of Chinese Academy of Sciences (Grant No.XDA27020200), the Youth Innovation Promotion Association CAS, and Yunnan Provincial Major Science and Technology Special Plan Projects (No.202202AD080004).

References


Bo Li, Wei Ye, Zhonghao Sheng, Rui Xie, Xiangyu Xi, and Shikun Zhang. 2020. Graph enhanced dual attention network for document-level relation extraction.


A Appendix

A.1 Training Strategy

In the *Relationship Classification* stage, previous work (Wang et al., 2019b) observed that there is an imbalance relation distribution for RE (the relational entity pairs are sparse). To alleviate the negative impact of sparsity, Zhang et al. (2021) introduces a balanced softmax method inspired by the circle loss (Sun et al., 2020). Based on this, we design **Adaptive Softmax** loss, which introduces a addition threshold class $\text{TH}$, which is automatically learned in the same way as other classes. The class TH aims to separate positive classes and negative classes, hoping that the scores of the target category are all greater than $s_{\text{TH}}$ and the scores of the non-target categories are all less than $s_{\text{TH}}$.

Formally,
\[
L_{rel} = \log(e^{r^T h} + \sum_{i \in \omega_{neg}} e^{s_i}) + \log(e^{-s^T h} + \sum_{i \in \omega_{pos}} e^{-s_i})
\]  

(9)

In the **Potential Pair Prediction** stage, in order to match binary classification task, we design the loss as:

\[
L_{pot} = -\frac{1}{n_p} \sum_{i=1}^{n_p} (y_i \log P_{pair} + (1-y_i) \log(1-P_{pair}))
\]  

(10)

where \(n_p\) is the size of full entity pairs set. In the **Cross Matching** stage, to capture the features of entities as subject and object respectively, we design the loss as:

\[
L_{sub} = -\frac{1}{n_c n_e} \sum_{j=1}^{n_c} \sum_{i=1}^{n_e} (y_i \log F_{sub}^j + (1-y_i) \log(1-F_{sub}^j))
\]  

\[
L_{obj} = -\frac{1}{n_c n_e} \sum_{j=1}^{n_c} \sum_{i=1}^{n_e} (y_i \log F_{obj}^j + (1-y_i) \log(1-F_{obj}^j))
\]  

(11)

where \(n_c\) is the size of full relation set, \(n_e\) is size of full entities set. The total loss is the sum of the above losses:

\[
L_{total} = \alpha L_{rel} + \beta L_{pot} + \gamma \frac{L_{sub}+L_{obj}}{2}
\]  

(12)

Performance might be better by carefully tuning the weight of each sub-loss, but we just assign equal weights for simplicity (ie., \(\alpha = \beta = \gamma = 1\)).