Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2022.028352
Article

Tech Science Press

Attention Weight is Indispensable in Joint Entity and Relation Extraction

Jianquan Ouyang[1,*], Jing Zhang[1] and Tianming Liu[2]

[1]Key Laboratory of Intelligen Computing and Information Processing, Ministry of Education, Computer science College of Xiangtan University, Xiangtan, 411100, China
[2]Department of Computer Science, University of Georgia, Athens, USA
*Corresponding Author: Jianquan Ouyang. Email: oyjq@xtu.edu.cn
Received: 08 February 2022; Accepted: 11 April 2022

Abstract: Joint entity and relation extraction (JERE) is an important foundation for unstructured knowledge extraction in natural language processing (NLP). Thus, designing efficient algorithms for it has become a vital task. Although existing methods can efficiently extract entities and relations, their performance should be improved. In this paper, we propose a novel model called Attention and Span-based Entity and Relation Transformer (ASpERT) for JERE. First, differing from the traditional approach that only considers the last hidden layer as the feature embedding, ASpERT concatenates the attention head information of each layer with the information of the last hidden layer by using an attentional contribution degree algorithm, so as to remain the key information of the original sentence in a deep transferring of the pre-trained model. Second, considering the unstable performance of the linear span classification and width embedding structure of the SpERT, ASpERT uses a multilayer perceptron (MLP) and softmax-based span classification structure. Ablation experiments on the feature embedding and span classification structures both show better performances than SpERT's. Moreover, the proposed model achieved desired results on three widely-used domain datasets (SciERC, CoNLL04, and ADE) and outperforms the current state-of-the-art model on SciERC. Specifically, the F1 score on SciERC is 52.30%, that on CoNLL04 is 71.66%, and that on ADE is 82.76%.

Keywords: Attentional contribution degree; joint entity and relation extraction; BERT; span

1 Introduction

Entity and relation extraction (ERE) has received much attention as a fundamental task in NLP, especially in specific domains (e.g., science, journalism, and medicine). The purpose of ERE is to extract structured triplets automatically from unstructured or semistructured natural language texts. A triplet consists of two entities and the relationship between them, and a sentence may contain multiple triplets. Owing to nested entities and overlapping relations, the extracted triplets may have similar or identical entities, and a triplet itself may contain two identical entities (with different relationships).

ERE is divided into pipeline ERE [1] and joint ERE (JERE) [2–5]. Their difference is the execution sequence of two subtasks, named entity recognition (NER) [6–9] and relation extraction (RE) [10]. Specifically, pipeline ERE first extracts entities from the text and then extracts relations between every two entities. In this serial execution, the success of RE most likely depends on the results of NER, and the lack of information interaction between NER and RE can cause errors to accumulate. Compared with the pipeline method, the joint method uses a parameter sharing or joint decoding mechanism between NER and RE. Such a mechanism enhances the information interaction between NER and RE, reduces the high dependence of RE on NER results, and improves the accuracy of ERE. JERE includes three directions: tagging [11], table filling [12], and sequence to sequence (Seq2Seq) [13]. Studies are considerably inclined to methods based on BIO/BILOU labels, and some complex algorithms may cause unbearable computational costs. Unlike BIO/BILOU labels, span-based methods [14] can efficiently identify nested entities, such as “phenytoin” within “phenytoin toxicity.”

Known as state-of-the-art span-based JERE, Span-based Entity and Relation Transformer (SpERT) [15] uses a sufficient number of strong negative samples and localized context to construct lightweight inference of BERT [16] embeddings, but this model still has two main flaws. First, SpERT focuses on learning span representation and lacks clear boundary supervision of entities. That is, the model relies on a width embedding layer to train the span length and directly classifies the sampled span through a fully connected layer. Second, many BERT-based JERE models (including SpERT) do not fully exploit domain-specific information. The semantic learning of sentences by using these models mainly comes from the coding information of the last hidden layer obtained through fine-tuning the BERT model, which limits the model’s performance.

To solve the problems mentioned above, we propose Attention and Span-based Entity and Relation Transformer (ASpERT), which is a JERE model based on the attentional contribution degree and MLP-softmax span classification structure. In ASpERT, a more complex MLP is added to enhance the entity boundary detection. In addition, in JERE’s studies on Transformer [17,18], the multihead self-attention is used to capture interactions among tokens, but only the last hidden layer is considered as the feature embedding for downstream tasks. In this paper, we develop a novel attentional contribution degree algorithm, which concatenates the softmax score of the attention head and the hidden layer feature embedding. View as a training strategy, this algorithm remains the strong attention between words by backpropagating to learn query vectors and key vectors in the pre-trained model. Finally, weighted joint optimization of the multitask loss function is conducted in the training process.

ASpERT is compared with state-of-the-art methods on three datasets, SciERC, CoNLL04 and ADE (public dataset repository address: http://lavis.cs.hs-rm.de/storage/spert/public/datasets/). Specifically, our model shows a significant performance improvement with a 1.39% increase in F1 score comparing to the baseline model (SpERT). Our model outperforms the current state-of-the-art model on the SciERC dataset and achieves desired results on CoNLL04 and ADE. In addition, we also investigate how to set contribution thresholds and different fusion methods more efficiently. And in the ablation experiments, we demonstrate the effectiveness of the novel span classification structure and attentional contribution degree algorithm.

The contributions of our work can be summarized as follows:

a) We analyze the reasons for the inaccurate boundary recognition of SpERT and propose a simple and effective span classification structure to alleviate this problem.

b) We propose an attentional contribution degree algorithm to enhance the model with strong attention between words by backpropagation.

c) Experiments show that our model achieves outstanding performance on domain-specific datasets (SciERC, CoNLL04, and ADE) in science, news, and medicine. Especially, it is better than the current state of the art on SciERC.

2 Related Work

Acting as an implementation of ERE, the pipeline method [19,20] executes NER and RE in series. Herein, NER methods [21] can be categorized into rule, dictionary, and machine learning-based methods [22–24]. ER methods can be divided into handcrafted feature-based methods [25] and neural network-based methods [26–28]. Although the pipeline method has been successfully applied in some fields, the sequential execution of NER and RE makes it ignore the correlation between the two tasks, which limits the further development of these methods.

To alleviate the above limitations, researchers proposed JERE, including feature-based methods [29,30] and neural network-based methods [31–35]. Limited by the expression capability of the model, later studies are mainly based on the neural network method. Research on JERE includes three main directions: tagging, table filling, and Seq2Seq. Zheng et al. [36] proposed a novel tagging scheme, which assigns a tag to each word (including word position, relation type, and relation role) for classification. The table filling [37] is usually to construct a two-dimensional table; thus, the solutions of NER and RE become the problems of labeling diagonal and nondiagonal elements in the table, respectively. These methods allow a single model to execute NER and RE simultaneously but cannot fully use the table structure. Wang et al. [38] proposed to learn two separate encoders (a table encoder and a sequence encoder), which effectively alleviates this problem. The Seq2Seq method [39] first retains sentence features and then extracts triplets in sequence. CopyRE [40], the most typical method, is based on the copy mechanism and Seq2Seq structure, but only extracts individual word. In response to this problem, Zeng et al. [41] proposed a multitask learning method based on BIO labeling.

Methods aforementioned are all based on the BIO/BILOU scheme, and they face a common problem—nested entities. To solve the problem, Takanobu et al. [42] adopted a hierarchical reinforcement learning framework. In this framework, entities and relations are divided into different levels, and the semantic information detected by high-level relations is used in extracting low-level entities. The two levels alternate back and forth to achieve JERE. Dai et al. [43] proposed a position-attention mechanism to solve this problem. It uses tag sequences that have the same length as the sentence to annotate each word. Although these methods alleviate the nested entity problem, the immense computational burden is inevitable.

An alternative to the BIO/BILOU scheme is the span-based method [44], which performs a detailed search on all spans to prevent the interference of nested entities on JERE results. This method enhances the interaction among tasks by refining the span representation, allowing the model to learn useful information from a broader context. The methods include the bi-LSTM-based span-level model proposed by Dixit et al. [45] and the dynamic span graph approach through soft coreference and relation links proposed by Luan et al. [46]. To improve the performance of the span method further, Wadden et al. [47] replaced the BiLSTM encoder with Transformers and combined it with BERT encodings and graph propagation to capture context relevance. Recently, Eberts and Ulges' SpERT [15] found localized context representation and strong negative sampling to be of vital importance. Although SpERT is the state-of-the-art model for span-based JERE, it suffers from underutilization of BERT encoding information and inaccurate identification of span boundaries.

3 Background

In this section, we introduce the baseline model, SpERT. It uses pretrained BERT as the core, tokenizes the input sentence, and applies span classification, span filtering, and relation classification. Specifically, it classifies each span into entity types, filters nonentities, and categorizes all candidate entity pairs. To train the classifier efficiently, SpERT uses negative samples at the model training stage.

3.1 Negative Sampling

Negative sampling is performed on each sentence $d_i(d_i \in D)$ in *corpus D*. A fixed number of negative samples are randomly sampled from sentence d_i and labeled with *none*, which is combined with the positive samples of existing labels in *corpus D* to form training samples (including candidate span and candidate entity pair). Then, the training samples are applied in learning the span and relation classifiers. For the span classifier, SpERT selects subsequences that do not belong to the positive span sample and are less than 10 words as the negative span sample. For the relation classifier, SpERT selects entity pairs without any relation labels from positive span samples as negative relation samples (See supplementary file for details).

3.2 Span Classification

The span classifier of SpERT consists of a fully connected layer and a softmax layer, and regards any candidate span $s := (e_{i+1}, e_{i+2}, \ldots, e_{i+k})$ as input (where e_i represents the i-th token embedding). Its output is the entity class probability $\hat{y}^s$ of this candidate span (where $\circ$ denotes concatenation):

$$e(s) = MaxPooling(f_l(e_{i+1}, e_{i+2}, \ldots, e_{i+k})) \circ w_k \tag{1}$$

$$x^s = e(s) \circ C_{[cls]} \tag{2}$$

$$\hat{y}^s = softmax(W^s \cdot x^s + b^s) \tag{3}$$

where $f_l(e_{i+1}, e_{i+2}, \ldots, e_{i+k})$ is the last hidden layer embedding from the fine-tuned BERT. w_k is width embedding, which learns the width of each candidate span from a dedicated embedding matrix. *MaxPooling* is the maximum pooling. $C_{[cls]}$ is the last hidden layer embedding from BERT's special [CLS] token. $W^s \in \mathcal{R}^{(2d_l+n) \times k_s}$ is the trainable weight, and $b^s \in \mathcal{R}^{2d_l + n}$ is the bias. d_l is the dimension of the BERT's last hidden layer, n is the dimension of w_k, and k_s is the number of entity classes (including *none*) *softmax* is the softmax activation function.

3.3 Span Filtering

The entity classes include predefined entity types (Tab. 2) and *none* label that does not constitute entities. In accordance with the output of the span classifier (Eq. (3)), the entity class with the highest probability is selected as the predicted result. If the predicted probability of the *none* label is the largest, then the candidate span does not constitute an entity.

3.4 Relation Classification

The relation classifier consists of a fully connected layer and sigmoid. The input of the classifier is any candidate entity pair (s_1, s_2), and the output is the relation class probability $\hat{y}^r$ of this candidate entity pair:

$$x^r = e(s_1) \circ C_{(s_1,s_2)} \circ e(s_2) \tag{4}$$

$$\hat{y}^r = \sigma(W^r \cdot x^r + b^r) \tag{5}$$

where $e(s_1)$ and $e(s_2)$ are the BERT/width embedding (Eq. (1)) of the head entity s_1 and the tail entity s_2 in the candidate entity pair (s_1, s_2). $C_{(s_1,s_2)}$ is the localized context representation. Specifically, SpERT places the span between the head entity and the tail entity into the fine-tuned BERT for encoding and obtains $C_{(s_1,\, s_2)}$. If this span is empty, then Eq. (4) is changed to $e(s_1) \circ e(s_2)$. $W^r \in \mathcal{R}^{(3d_l + 2n) \times k_r}$ is the trainable weight, $b^r \in \mathcal{R}^{3d_l + 2n}$ is the bias, and k_r is the number of relation classes (including *none*). σ is the sigmoid activation function. Given a threshold α, any relation class probability greater than α is considered activated. If *none* is activated, then this entity pair has no known relation. For example, if the predicted probabilities of the entity pair $(s_1,\ s_2)$ with respect to $Adverse - Effect$, *Drug*, and *none*. are

0.43, 0.47, and 0.1, respectively, then there are two types of relationships between $Adverse - Effect$ and $Drug$. If the predicted probabilities are 0.59, 0.0, 0.41, respectively, then no relationship exists for that entity pair (The threshold α is set at 0.4).

3.5 Problems of SpERT

As mentioned in the Introduction, we determine that SpERT has two problems. First, SpERT's classifier lacks clear boundary supervision on the span. Width embedding is the only constraint mechanisin span width. Considering that the span is long or short, spans composed of different numbers of words will have distinct characteristics. SpERT specifically learns a width embedding matrix through backpropagation; hence, it should play a key role in entity boundary supervision. To evaluate the effectiveness of width embedding, we test two different training models on three datasets:

- SpERT: It uses the default structure settings, which provide the width embeddings that need to be learned by backpropagation (Eq. (1)).
- ERT': The variant model of SpERT that removes the width embedding in the span and relational classifiers, while keeping the other default structure settings of the model.

As shown in Tab. 1, the addition of width embedding is unreliable in improving the performance of the span classifier. Especially on the SciERC dataset, the F1 score of the SpERT model with width embedding decreases by 0.75% in terms of NER. Three reasons are considered for our analysis. First, the model lacks boundary supervision when facing a complex dataset. The SciERC dataset is more complicated than the two other datasets. It is more significant than CoNLL04 in the dataset size, and it is 3 times that of ADE in the entity class. Second, the width embedding of SpERT only learns the span width and cannot essentially solve the problem of inaccurate boundary recognition. Consequently, performance degradation is expected. Third, because the span classifier of SpERT is only a fully connected layer, the model is overly dependent on BERT encoding. For example, when the extraction target is the "geometric estimation problem," the model extracts the correct span while also extracting the semantically similar wrong span "selection of geometric estimation problems," which leads to a decrease in model performance.

Table 1: Results of SpERT and SpERT' on three datasets. The effectiveness of width embedding for the SpERT model is evaluated

Dataset	Model	Precision	Recall	F1
CoNLL04	SpERT	87.99	**89.62**	**88.80**
	SpERT'	**88.15**	88.97	88.56
SciERC	SpERT	69.34	68.84	69.09
	SpERT'	**70.01**	**69.67**	**69.84**
ADE	SpERT	**90.83**	91.18	**91.00**
	SpERT'	89.00	**91.37**	90.17

In addition, many experiments have shown that the BERT model effectively extracts text information. If the text data are domain-specific (e.g., science, news, and medicine), we may need to consider creating our domain-specific language model. Relevant models have been created by training the BERT architecture on a domain-specific *corpus* rather than the general English text *corpus* used to train the original BERT model. Because pretraining BERT requires a large *corpus*, and we cannot use this method to improve the model's extraction of text in a specific field. Therefore, we need to change the method to mine the unexploited information in the Transformers pretrained model under the existing conditions. At present, the input for downstream tasks of many mainstream models (including SpERT) often comes from the last hidden layer embedding of BERT while ignoring the interactive information among words carried by the BERT attention head itself. To this aim, we provide a novel attentional contribution degree algorithm, which combines the softmax attention head score with hidden layer feature embedding to improve the model's extraction of entities and relationships.

4 Our Method

In this section, we choose SpERT as the baseline model, analyze SpERT's problems of inaccurate span recognition and insufficient information mining in specific fields, and propose a novel ASpERT model (Fig. 1). Then, we introduce a novel attentional contribution degree algorithm and a multitask training method that combines span and relation classifiers.

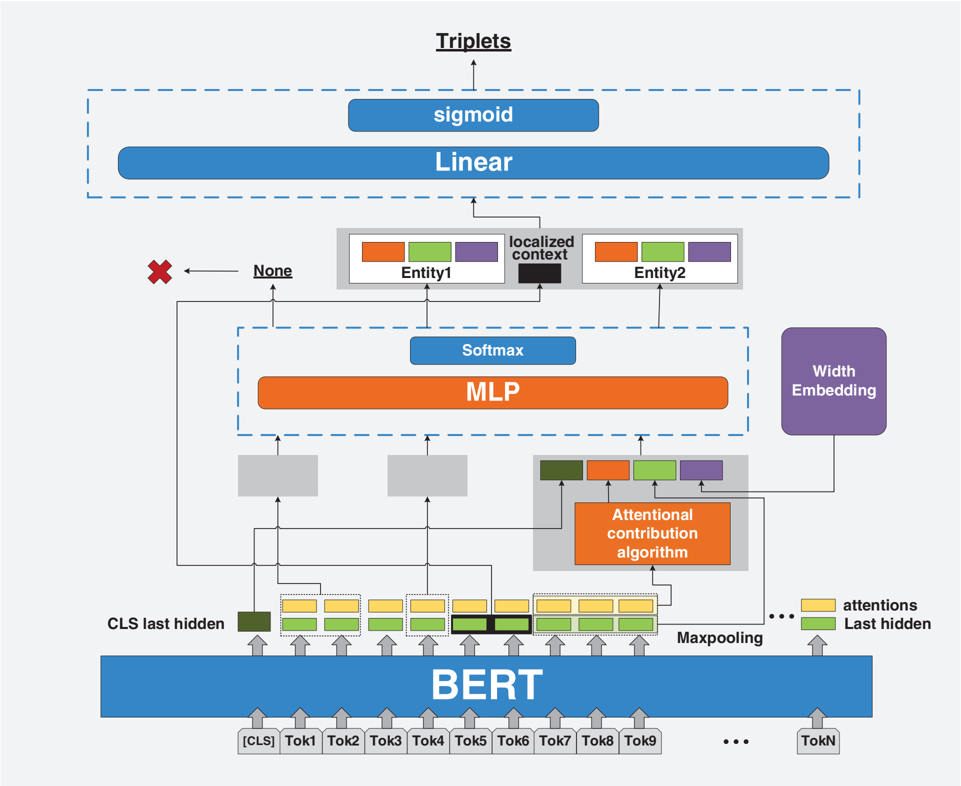

Figure 1: Overview of the ASpERT model for JERE and the orange part is our contribution

4.1 Novel Structure for the Span Classifier

We consider that the span classifier is different from the traditional classifier, such as fully connected layer and softmax layer. In addition to classifying the span, it also needs to predict which words belong to the entity boundary. Thus, we propose a span classification structure that considers these two functions.

The BERT embedding of candidate span $f_l(e_{i+1}, e_{i+2}, \ldots, e_{i+k})$ and the BERT embedding of the special [CLS] token $C_{[cls]}$ are the main sources of textual semantic information (Eq. (1)). The special [CLS] token represents the complete sentence information in the classification task. The maximum pooling of these BERT embeddings is only applicable to span classification. A candidate span includes one or more words, and BERT assigns an embedding matrix to each word through fine-tuning. Maximum pooling (Eq. (1)) of the BERT embedding of the candidate span is equivalent to selecting the largest single word embedding matrix to represent the semantic information of this span. SpERT excessively strengthens key information at the expense of marginal information and the association among words, resulting in the unclear boundary supervision of entities. This condition explains well why "geometric estimation problems" and "selection of geometric estimation problems" have similar span class probabilities. For the above reasons, we add the attentional contribution degree $f_a(e_{i+1}, e_{i+2}, \ldots, e_{i+k})$ to the span representation as the boundary confidence of span classification. Specifically, we first concatenate all the attention heads with the residuals of the span classifier, then remove the lower attention scores and take the average (*via* the attention contribution algorithm in Section 4.2) combined with the feature embedding. Finally, the fine-tuning of the pre-trained model is constrained by backpropagation learning. We consider that the attentional contribution degree incorporates word-to-word attention as well as residual concatenation, allowing the model not to lose original information as depth increases. The specific improvements to Eqs. (1) and (2) respectively are as follows:

$$E(s) = MaxPooling(f_l(e_{i+1}, e_{i+2}, \ldots, e_{i+k})) \circ f_a(e_{i+1}, e_{i+2}, \ldots, e_{i+k}) \circ w_k \tag{6}$$

$$X^s = E(s) \circ C_{[cls]} \tag{7}$$

where $f_l(e_{i+1}, e_{i+2}, \ldots, e_{i+k})$, w_k and $C_{[cls]}$ are obtained in the same way as in Eq. (1). The details of the calculation of $f_a(e_{i+1}, e_{i+2}, \ldots, e_{i+k})$ are described in the Attentional Contribution Degree Algorithm section.

JERE tasks are usually converted into one or more classification tasks at the end. Therefore, the classifier's quality is related to whether the high-dimensional data information can be accurately mapped to a given category. SpERT's span classifier is a linear fully connected layer. Few data strictly adhere to the linear distribution when noise is introduced, such that a simple linear structure cannot accurately predict the span class. Recently, MLP has been repositioned in visual classification [48]. For migration learning, we use MLP for span classification, hoping to increase the number of parameters to improve the potential representation capability of the classifier. The improvements to Eq. (3) are as follows:

$$Y^{s'} = ReLU(W^s{}_1 \cdot X^s + b^s{}_1) \tag{8}$$

$$\hat{Y}^s = softmax\left(W^s{}_2 \cdot Y^{s'} + b^s{}_2\right) \tag{9}$$

where $\hat{Y}^s$ is the entity probability. $W^s{}_1 \in \mathcal{R}^{(2d_l + n + d_a) \times m}$ and $W^s{}_2 \in \mathcal{R}^{m \times k_s}$ are the trainable weights. $b^s{}_1 \in \mathcal{R}^{(2d_l + n + d_a)}$ and $b^s{}_2 \in \mathcal{R}^m$ are the biases. d_a is the number of BERT's attention heads. m is the number of hidden layer units of the MLP. *ReLU* is the ReLu activation function, and *softmax* is the softmax activation function.

4.2 Attentional Contribution Degree Algorithm

In this subsection, we describe the attentional contribution degree algorithm in detail. Attentional contribution degree is a novel attention weight, which concatenates the calculated attentional contribution degree with hidden layer features to obtain a weighted feature encoding. This encoding helps the model understand the contextual information of the span and strengthens the model's extraction of entities and relations.

Algorithm 1: Mask score of entity sample

```
Input: The sentence, $D$; The entity contained in sentence, $s := (e_{i+1}, \ldots, e_{i+k})$;
Output: The mask score of entity sample, $Mask_s$;
for all $d$ such that $d \in D$ do
    if $d \in s$ then
        $Mask_s(d) \leftarrow -\infty$
    else
        $Mask_s(d) \leftarrow 0$
    end if
end for
return $Mask_s$
```

The attentional contribution degree is derived from the attention paid to interword information by each attention head in each layer of the pretrained model. Among them, pretrained model comes from the BERT variant of the Transformers library. The large model and *corpus* symbolize many GPU resources, such that we only fine-tune the pretrained model (such as BERT base (cased) [16], SciBERT (cased) [49], and BioBERT (cased) [50]) in a specific field. This condition does not mean that we are bound by the pretrained model. On the contrary, we fully utilize the attention header information of Transformers. We train and use the intermediate product of the model—self-attention head. For example, BERT base has 12 layers, and each layer has 12 attention heads. Then we can make use of the information of these 144 attention heads.

Specifically, first we extract all the attention heads, which contain information about the relationship among words in a sentence. Second, we concatenate multiple attention heads in the num head dimension. As shown in Algorithm 1, we mask irrelevant words and only retain the relationship information between the candidate span and the words in the full text. Immediately after, considering that each attention layer provides multiple "representation subspaces," the multihead attention mechanism expands the model's ability to represent different positions. We provide the contribution threshold θ to filter the attention head information with low attention to candidate span. Finally, the attention contribution degree is obtained by mean-pooling the attention header information from the token dimensions of both the context and the entity. (Algorithm 2).

Algorithm 2: Attentional contribution degree

```
Input: The entity sample, $s := (e_{i+1}, \ldots, e_{i+k})$; The mask score of entity sample, $Mask_s$; The Bert model pretrained with domain-specific datasets, $M_s$; Contribution degree threshold, $\theta$; A method to find the mean value from the token dimension of the context and the token dimension of the entity, $MeanPooling$;
Output: The attentional contribution degree, $f_a(s)$;
$A^{s'} \leftarrow M_s[attentions]$
$A^s \leftarrow A^{s'} + Mask_s$
$A_{temp} \leftarrow A^s$
for all $a$, $b$ such that $a \in A^s$, $b \in A_{temp}$ do
    if $a > \theta$ then
        $b \leftarrow 1$
```

(Continued)

Algorithm 2: (continued)

```
    else
        $b \leftarrow 0$
    end if
end for
$A^s \leftarrow A^s \cdot A_{temp}$
$f_a(s) \leftarrow MeanPooling(A^s)$.
return $f_a(s)$
```

4.3 Training

Our training is supervised, providing the model with labeled sentences (including candidate span, entity class, candidate entity pair, and relation class). We learn width embedding w_k and span/relation classifiers' parameters ($W^s{}_{(\cdot)}$, $b^s{}_{(\cdot)}$, W^r, b^r) and fine-tune the domain-specific BERT. Different from the joint loss function defined by SpERT, Eq. (10) is used here for entity classification and relation classification:

$$L = \lambda L^s + L^r \tag{10}$$

where λ is the weight of the joint loss function, L^s is the loss of the span classifier calculated using the cross-entropy loss function, and L^r is the loss of the relation classifier calculated using the binary cross-entropy loss function.

5 Experiment

5.1 Datasets and Setting

We evaluate the model on three datasets from different domains, CoNLL04 [51], SciERC [14], and ADE [52]. As shown in Tab. 2, the CoNLL04 dataset is derived from news articles and includes four entity types and five relationship types. The dataset is divided into a training set of 911 sentences, a validation set of 231 sentences, and a test set of 288 sentences. The SciERC (scientific information extractor) dataset is derived from abstracts of artificial intelligence papers and includes six scientific entity types and seven relationship types. This dataset is divided into a training set of 1861 sentences, a validation set of 275 sentences, and a test set of 551 sentences. The ADE (adverse drug effect) dataset is derived from medical reports describing the adverse effects of drug use and contains two entity types and one relationship type. The dataset is divided into a training set of 3843 sentences and a validation set of 429 sentences.

Table 2: Detailed information of datasets

Dataset	Entity type	Relationship type
CoNLL04	Location, organization, people, other	Work-for, kill, organizationbased-in, live-in, located-in
SciERC	Task, method, metric, material, other-scientific-term, generic	Compare, conjunction, evaluate-for, used-for, featureof, part-of, hyponym-of
ADE	Adverse-effect	Adverse-effect, drug

We evaluated ASpERT on entity extraction and RE. An entity prediction is considered correct if the span and entity type of the entity prediction match the ground truth. A relation prediction is considered correct if the relation type and the two related entities (span and type) match the ground truth. In particular, to be consistent with the evaluation criteria of the comparative model, we only consider the prediction of relationship and entity span (ignoring the accuracy of entity type) on the SciERC dataset. Hyperparameters used for final training are listed in Tab. 3.

Table 3: Optimal hyperparameters used for final training on the ADE, SciERC, and CoNLL04 datasets

Hyperparamete	CoNLL04	SciERC	ADE
Entity negative sampling number	150/sentence	150/sentence	150/sentence
Relation negative sampling number	150/sentence	150/sentence	150/sentence
Pre-trained model type	BERT base (cased)	SciBERT (cased)	BioBERT (cased)
Span classifier MLP size m	784	784	784
width embedding size n	25	25	25
Contribution threshold θ	0.5	0.5	0.5
Relation classifier threshold α	0.4	0.4	0.4
MLP dropout	0.1	0.1	0.1
Optimizer	Adam	Adam	Adam
Peak learning rate	5e−5	5e−5	5e−5
Linear warmup learning rate	0.1	0.1	0.1
Linear decay learning rate	0.01	0.01	0.01
Epochs	20	20	20
Batch size	4	6	10

5.2 Comparison with the State of the Art

First, to evaluate the effectiveness of ASpERT's improvement based on SpERT, we train both models on the same device and unify the pretrained model and training parameters. We report an average of over five runs for each dataset. In particular, the ADE dataset uses 10-fold cross validation. As shown in Tab. 4, the performance of ASpERT is significantly better than that of the baseline model (SpERT) on different datasets. For entity extraction, the micro-F1 scores are increased by 0.45% (CoNLL04), 0.20% (SciERC), and 0.52% (ADE), and the macro-F1 scores are increased by 0.71% (CoNLL04), 0.33% (SciERC), and 0.50% (ADE). For RE, the micro-F1 scores are increased by 1.25% (CoNLL04), 1.39% (SciERC), and 1.31% (ADE), and the macro-F1 scores are increased by 1.25% (CoNLL04), 1.29% (SciERC), and 1.31% (ADE).

Subsequently, we compared the proposed model with the most advanced models currently. As shown in Tab. 5, these models are the top four models (except for SpERT) of the three datasets in the Papers With Code ranking list. We sorted ASpERT and these models in descending order in accordance with the F1 score of RE. The experimental results show that ASpERT has higher extraction performance in entities and relations. Even in the challenging and domain-specific SciERC dataset, ASpERT's F1 score RE is 0.30% higher than that of the top-ranked PL-Marker.

Table 4: Results of SpERT and ASpERT on three datasets (metrics: microaverage = †, macroaverage = ‡)

Dataset	Model	Entity			Relation		
		Precision	Recall	F1	Precision	Recall	F1
CoNLL04	SpERT†	87.64	89.03	88.32	70.72	67.58	69.11
	SpERT‡	84.75	85.86	85.26	72.11	69.24	70.41
	ASpERT†	89.03	88.53	88.77	73.62	67.39	70.36
	ASpERT‡	86.57	85.49	85.97	74.92	69.01	71.66
SciERC	SpERT†	69.46	70.59	70.02	52.20	49.71	50.91
	SpERT‡	69.28	70.62	69.79	51.74	47.87	49.23
	ASpERT†	69.88	70.57	70.22	53.59	51.07	52.30
	ASpERT‡	69.72	70.67	70.12	53.26	48.88	50.52
ADE	SpERT†	89.83	91.40	90.60	79.70	83.29	81.45
	SpERT‡	90.10	91.74	90.91	79.70	83.29	81.45
	ASpERT†	90.68	91.56	91.12	81.65	83.92	82.76
	ASpERT‡	90.96	91.87	91.41	81.65	83.92	82.76

Table 5: Results of comparing ASpERT with the state-of-the-art models on three datasets (metrics: microaverage = †, macroaverage = ‡)

Dataset	Model	Entity	Relation
CoNLL04	REBEL [39]‡	–	76.65
	Table-Sequence [38]‡	86.90	75.40
	Deeper [3] ‡	87.00	72.63
	ASpERT‡	**85.97**	**71.66**
	Biaffine attention [2]‡	86.20	64.40
SciERC	**ASpERT†**	**70.22**	**52.30**
	PL-Marker [35]†	69.90	52.00
	SpERT.PL [33]†	70.53	51.25
	Ours: cross-sentence [22]†	68.90	50.10
	DyGIE++ [47]†	67.50	48.40
ADE	Deeper [3] ‡	89.48	83.74
	PFN [32] ‡	91.30	83.20
	ASpERT‡	**91.41**	**82.76**
	REBEL [39]‡	–	82.20
	CMAN [34]‡	89.40	81.14

5.3 *Effects of Attentional Contribution Degree*

In Tab. 4, although the performance of ASpERT is better than that of SpERT, it is still not clear which part of ASpERT plays a key role. To demonstrate the advantage of the attentional contribution algorithm in JERE, we test two models:

- Full: We use the complete ASpERT model structure.
- -AC: We retain most of the ASpERT model structure but remove the attentional contribution degree algorithm.

We ran these two models more than 5 times on three datasets and average them (the ADE dataset uses 10-fold cross validation). As shown in Tab. 6, the performance of the variant model without the attentional contribution degree algorithm is significantly decreased. In terms of entity extraction, F1 scores decreased by 0.48%. In RE, the F1 score decreased by 1.46%. These experimental results show that the attentional contribution degree algorithm can capture word-to-word relationships adequately, which helps in efficient relation classification and is the main contribution of the new model architecture.

Table 6: Effect of attentional contribution degree algorithm on ERE

Dataset	Model	Entity F1	Relation F1
ConLL04	Full	85.97	71.66
	-AC	86.10	70.20
SciERC	Full	70.22	52.30
	-AC	70.05	51.56
ADE	Full	91.41	82.76
	-AC	90.93	81.73

Then, we investigated the effect of setting different contribution thresholds on the model's ability to capture word-to-word relationships on SciERC and CoNLL04. Fig. 2 shows the F1 scores (RE) with different contribution thresholds. When the threshold is 0.5, the model performance is optimal.

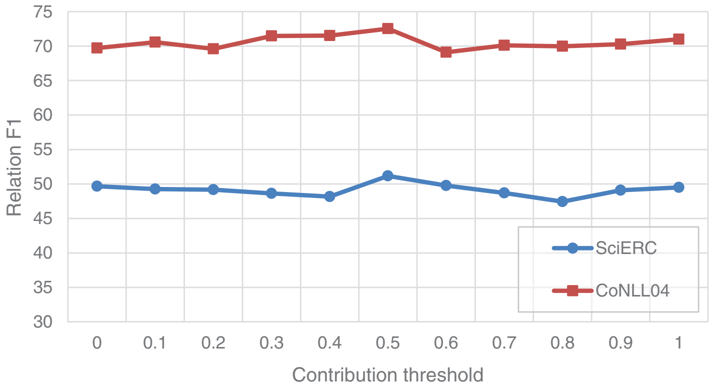

Figure 2: Effect of different contribution thresholds on the relation F1 score

Lastly, we also investigate the different fusion methods of each attention head information, namely, the maximum pooling, sum pooling, and mean pooling. Tab. 7 shows the F1 scores by using different fusion methods on three datasets. We determined that the mean pooling is more advantageous for JERE.

Table 7: Study of different fusion methods for each attention head information

Dataset	Method	Entity F1	Relation F1
ConLL04	Max	85.68	69.17
	Sum	86.38	70.83
	Mean	86.8	71.21
SciERC	Max	70.10	51.16
	Sum	70.21	51.79
	Mean	69.80	52.38
ADE	Max	91.27	82.34
	Sum	91.73	82.09
	Mean	91.96	82.94

5.4 Effects of the Novel Span Classifier

To evaluate the effectiveness of the novel span classifier, we further test two models on the SciERC dataset:

- Full: We use the complete ASpERT model structure.
- -MLP: We retain most of the ASpERT model structure but replace the MLP structure with a fully connected layer in span classification.

As shown in Tab. 8, removing the MLP structure weakened the classifier's ability to learn information about span boundaries, leading to a decrease in the recall and accuracy of entity extraction and thus a decrease in the F1 score by nearly 0.74%.

Table 8: Effect of MLP on entity extraction

Dataset	Model	Entity Precision	Entity Recall	Entity F1
ConLL04	Full	86.57	85.49	85.97
	-MLP	84.84	85.71	85.23
SciERC	Full	69.88	70.57	70.22
	-MLP	69.74	70.49	70.11
ADE	Full	90.96	91.87	91.41
	-MLP	90.24	91.98	91.10

6 Conclusion

In this paper, we have proposed a novel model termed ASpERT for JERE. This model fuses the overlooked attention header information in downstream tasks with the feature embedding of the hidden layer *via* a new attentional contribution degree algorithm. Specifically, the attentional contribution incorporates word-to-word attention and the residual connectivity of the span classifier with each attentional head. This allows the model to maintain the raw information as depth increases and thus enhance the model's ability to capture contextual information, thus being adapted to domain-specific JERE. Moreover, the MLP-softmax structure of the span classifier and the attentional contributions is used to determine the boundary supervision and to improve the span classification. Without these ideas, researchers who are limited by hardware conditions may have to fine-tune parameters for information extraction tasks. The use of pre-trained models is not limited to the encoding of implicit layer information.

Considering that the attentional head is the base unit of Transformer pre-training models, in future work, we will further demonstrate the influence of the attentional contribution degree algorithm on other Transformer pre-training models. Notably, Asian languages, however, require more words to express the same meaning as English, which is not friendly to the random sampling method, hence we will focus on spanwise sampling of complex language structures.

Acknowledgement: We thank the open-source authors of the dataset. We also thank all members from Xiangtan University 504 Lab for their strong support for my research.

Funding Statement: This work was supported by Key Projects of the Ministry of Science and Technology of the People's Republic of China (2020YFC0832401) and National College Students Innovation and Entrepreneurship Training Program (No. 202110530001).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. R. Gormley, M. Yu and M. Dredze, "Improved relation extraction with feature-rich compositional embedding models," 2015. [Online]. Available: https://arxiv.org/abs/1505.02419.

[2] D. Q. Nguyen and K. Verspoor, "End-to-end neural relation extraction using deep biaffine attention," in *European Conf. on Information Retrieval*, Cologne, Germany, pp. 729–738, 2019.

[3] P. Crone, "Deeper task-specificity improves joint entity and relation extraction," 2002. [Online]. Available: https://arxiv.org/abs/2002.06424.

[4] C. Chen and F. Kong, "Enhancing entity boundary detection for better chinese named entity recognition," in *Proc. of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th Int. Joint Conf. on Natural Language Processing*, vol. 2, pp. 20–25, 2021.

[5] S. Zhao, M. Hu, Z. Cai and F. Liu, "Dynamic modeling cross-modal interactions in two-phase prediction for entity-relation extraction," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–10, 2021.

[6] G. Luo, X. Huang, C.-Y. Lin and Z. Nie, "Joint entity recognition and disambiguation," in *Proc. of the 2015 Conf. on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp. 879–888, 2015.

[7] Z. Liu and X. Chen, "Research on relation extraction of named entity on social media in smart cities," *Soft Computing*, vol. 24, no. 15, pp. 11135–11147, 2020.

[8] C. Tan, W. Qiu, M. Chen, R. Wang and F. Huang, "Boundary enhanced neural span classification for nested named entity recognition," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 5, pp. 9016–9902, 2020.

[9] J. Cheng, J. Liu, X. Xu, D. Xia, L. Liu *et al.,* "A review of chinese named entity recognition," *KSII Transactions on Internet and Information Systems*, vol. 15, no. 6, pp. 2012–2030, 2021.

[10] G. Zhou, J. Su, J. Zhang and M. Zhang, "Exploring various knowledge in relation extraction," in *Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics*, Ann Arbor, Michigan, USA, pp. 427–443, 2005.

[11] Y. Wang, B. Yu, Y. Zhang, T. Liu, H. Zhu *et al.,* "Tplinker: Single-stage joint extraction of entities and relations through token pair linking," 2010. [Online]. Available: https://arxiv.org/abs/2010.13415.

[12] M. Zhang, Y. Zhang and G. Fu, "End-to-end neural relation extraction with global optimization," in *Proc. of the 2017 Conf. on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, pp. 1730–1740, 2017.

[13] T. Nayak and H. T. Ng, "Effective modeling of encoder-decoder architecture for joint entity and relation extraction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, pp. 8528–8535, 2020.

[14] Y. Luan, L. He, M. Ostendorf and H. Hajishirzi, "Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction," 2018. [Online]. Available: https://arxiv.org/abs/1808.09602.

[15] M. Eberts and A. Ulges, "Span-based joint entity and relation extraction with transformer pre-training," 2019. [Online]. Available: https://arxiv.org/abs/1909.07755.

[16] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018. [Online]. Available: https://arxiv.org/abs/1810.04805.

[17] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma *et al.,* "Albert: A lite bert for self-supervised learning of language representations," 2019. [Online]. Available: https://arxiv.org/abs/1909.11942.

[18] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi *et al.,* "Roberta: A robustly optimized bert pretraining approach," 2019. [Online]. Available: https://arxiv.org/abs/1907.11692.

[19] M. Mintz, S. Bills, R. Snow and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proc. of the Joint Conf. of the 47th Annual Meeting of the ACL and the 4th Int. Joint Conf. on Natural Language Processing of the AFNLP*, Suntec, Singapore, pp. 1003–1011, 2009.

[20] Y. S. Chan and D. Roth, "Exploiting syntactico-semantic structures for relation extraction," in *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, pp. 551–560, 2011.

[21] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[22] Z. Zhong and D. Chen, "A frustratingly easy approach for entity and relation extraction," 2020. [Online]. Available: https://arxiv.org/abs/2010.12812.

[23] P. Chen, H. Ding, J. Araki and R. Huang, "Explicitly capturing relations between entity mentions via graph neural networks for domain-specific named entity recognition," in *Proc. of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th Int. Joint Conf. on Natural Language Processing*, Bangkok, Thailand, vol. 2, pp. 735–742, 2021.

[24] N. Alsaaran and M. Alrabiah, "Arabic named entity recognition: A bert-bgru approach," *Computers Materials & Continua*, vol. 68, no. 1, pp. 471–485, 2021.

[25] B. Rink and S. Harabagiu, "Utd: Classifying semantic relations by combining lexical and semantic resources," in *Proc. of the 5th Int. Workshop on Semantic Evaluation*, Uppsala, Sweden, pp. 256–259, 2010.

[26] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng *et al.,* "Classifying relations via long short term memory networks along shortest dependency paths," in *Proc. of the 2015 Conf. on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp. 1785–1794, 2015.

[27] S. Zheng, J. Xu, P. Zhou, H. Bao, Z. Qi *et al.,* "A neural network framework for relation extraction: Learning entity semantic and relation pattern," *Knowledge-Based Systems*, vol. 114, no. 8, pp. 12–23, 2016.

[28] Q. Yue, X. Li and D. Li, "Chinese relation extraction on forestry knowledge graph construction," *Computer Systems Science and Engineering*, vol. 37, no. 3, pp. 423–442, 2021.

[29] X. Yu and W. Lam, "Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach," in *Coling 2010: Posters*, Beijing, China: Coling 2010 Organizing Committee, pp. 1399–1407, 2010.

[30] M. Miwa and Y. Sasaki, "Modeling joint entity and relation extraction with table representation," in *Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1858–1869, 2014.

[31] M. Miwa and M. Bansal, “End-to-end relation extraction using lstms on sequences and tree structures,” 2016. [Online]. Available: https://arxiv.org/abs/1601.00770.

[32] Z. Yan, C. Zhang, J. Fu, Q. Zhang and Z. Wei, “A partition filter network for joint entity and relation extraction,” 2021. [Online]. Available: https://arxiv.org/abs/2108.12202.

[33] T. Santosh, P. Chakraborty, S. Dutta, D. K. Sanyal and P. P. Das, “Joint entity and relation extraction from scientific documents: Role of linguistic information and entity types,” in *Proc. of the 2nd Workshop on Extraction and Evaluation of Knowledge Entities from Scientific Documents (EEKE2021)*, Virtual Event, 2021.

[34] S. Zhao, M. Hu, Z. Cai and F. Liu, “Modeling dense cross-modal interactions for joint entity-relation extraction,” in *Proc. of the Twenty-Ninth Int. Conf. on International Joint Conf. on Artificial Intelligence*, Montreal-themed virtual reality, pp. 4032–4038, 2021.

[35] D. Ye, Y. Lin and M. Sun, “Pack together: Entity and relation extraction with levitated marker,” 2021. [Online]. Available: https://arxiv.org/abs/2109.06067.

[36] S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou *et al.,* “Joint extraction of entities and relations based on a novel tagging scheme,” 2017. [Online]. Available: https://arxiv.org/abs/1706.05075.

[37] P. Gupta, H. Schütze and B. Andrassy, “Table filling multi-task recurrent neural network for joint entity and relation extraction,” in *Proc. of COLING 2016, the 26th Int. Conf. on Computational Linguistics: Technical Papers*, Osaka, Japan, pp. 2537–2547, 2016.

[38] J. Wang and W. Lu, “Two are better than one: Joint entity and relation extraction with table-sequence encoders,” 2020. [Online]. Available: https://arxiv.org/abs/2010.03851.

[39] P.-L. H. Cabot and R. Navigli, “Rebel: Relation extraction by end-to-end language generation,” in *Findings of the Association for Computational Linguistics: EMNLP 2021*, Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 2370–2381, 2021.

[40] X. Zeng, D. Zeng, S. He, K. Liu and J. Zhao, “Extracting relational facts by an end-to-end neural model with copy mechanism,” in *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia, vol. 1, pp. 506–514, 2018.

[41] D. Zeng, H. Zhang and Q. Liu, “Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning,” *Proceedings of the AAAI Conference on Artificial Intelligence*, Hilton New York Midtown, New York, New York, USA, pp. 9507–9514, 2020.

[42] R. Takanobu, T. Zhang, J. Liu and M. Huang, “A hierarchical framework for relation extraction with reinforcement learning,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 7072–7079, 2019.

[43] D. Dai, X. Xiao, Y. Lyu, S. Dou, Q. She *et al.,* “Joint extraction of entities and overlapping relations using position-attentive sequence labeling,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 6300–6308, 2019.

[44] K. Ding, S. Liu, Y. Zhang, H. Zhang, X. Zhang *et al.,* “A knowledge-enriched and span-based network for joint entity and relation extraction,” *Computers Materials & Continua*, vol. 68, no. 1, pp. 377–389, 2021.

[45] K. Dixit and Y. Al-Onaizan, “Span-level model for relation extraction,” in *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, pp. 5308–5314, 2019.

[46] Y. Luan, D. Wadden, L. He, A. Shah, M. Ostendorf *et al.,* “A general framework for information extraction using dynamic span graphs,” 2019. [Online]. Available: https://arxiv.org/abs/1904.03296.

[47] D. Wadden, U. Wennberg, Y. Luan and H. Hajishirzi, “Entity, relation, and event extraction with contextualized span representations,” 2019. [Online]. Available: https://arxiv.org/abs/1909.03546.

[48] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai *et al.,* Mlp-mixer: An all-mlp architecture for vision. In: *Advances in Neural Information Processing Systems*. Virtual, Vol. 34, 2021.

[49] I. Beltagy, K. Lo and A. Cohan, “Scibert: A pretrained language model for scientific text,” in *Proc. of the 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int. Joint Conf. on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, 2019.

[50] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim *et al.,* "Biobert: A pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[51] D. Roth and W.-t Yih, "A linear programming formulation for global inference in natural language tasks," Illinois Univ at Urbana-Champaign Dept of Computer Science, 2004.

[52] H. Gurulingappa, A. M. Rajput, A. Roberts, J. Fluck, M. Hofmann-Apitius *et al.,* "Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports," *Journal of Biomedical Informatics*, vol. 45, no. 5, pp. 885–892, 2012.