Joint Extraction of Entities and Relations Based on Multi-feature Fusion

Shoubin Li [1 2] ,Zhiyuan Chang [2] and Yangyang Liu [3]
[1]University of Chinese Academy of Sciences, Beijing, China
[2]The Institute of Software, Chinese Academy of Sciences, Beijing, China
[3]University of Auckland, Auckland, New Zealand
{shoubin zhiyuan}@iscas.ac.cn, yilu660@aucklanduni.ac.nz

***Abstract*—Joint extraction of entities and relations is essential for understanding massive text corpora. In recent years, the span-based joint models have achieved excellent results in the entity and relation extraction task. However, the previous literature and experimental results suggest that the usage of span-based method in entity and relation extraction may produce more redundant entities, although it can solve the overlapping problem of entities. In order to solve the problem of entity redundancy, this paper proposes a joint extraction model based on multi-feature fusion. The overall network follows the framework as SpERT, which is the state-of-the-art model for joint entity and relation extraction. In addition to the word embedding features in SpERT, the proposed model also considers the part-of-speech features. We believe that the part-of-speech features in entities are helpful for entity recognition and can effectively alleviate the entity redundancy problem. The proposed model is evaluated on two public data sets, CoNLL04 and ADE. The experimental results show that the proposed joint extraction model based on multi-feature fusion significantly outperforms current state-of-the-art methods.**

***Index Terms*—Joint Entity and Relation Extraction, Span-based Method, Entity Redundancy, Pre-trained Model, Part-of-speech**

I. Introduction

Entity and Relation Extraction (ERE) is one of the critical tasks in information extraction [1]. There are two mainstream methods to solve the ERE task, the traditional pipelined method and the joint extraction method [2]. The traditional pipelined method divides the task into two independent sub-tasks, that is, the named entity recognition (NER) task [3], and the relation classification task [4], and design methods to solve these two sub-tasks respectively. Although the traditional method can make each sub-task more flexible, the performance of the NER task can play a decisive role in the whole model, which directly affects the results of relation extraction and even produces erroneous accumulation. Compared with the traditional pipelined methods, the joint extraction method solves these two tasks with a single model. As parameter sharing is realized in entity recognition and relation classification tasks under the joint extraction method, erroneous accumulation can be avoided.

In the newly proposed joint extraction models, the commonly-used tagging scheme is the sequence labeling based on the BIO/BILOU tags [5] [6]. However, the BIO tagging scheme, which divides the text in order, does not recognize the nested entities [7]. Recently, some researchers have proposed a span-based method [8] to extract entities and relations jointly. The span-based method treats a token of any length as an entity to identify overlapping entities such as ”apple pie” and ”apple”.

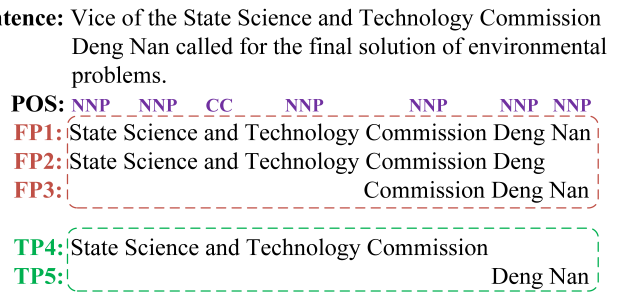

Fig. 1. An example of entity extraction. POS is the abbreviation for part-of-speech, corresponding to the words below. FP is short for false positive, and TP denotes true positive. Entities contained in the red block are the ones with the wrong extraction, and those in the green block are the ones with the correct extraction.

With the development of pre-trained model on a small amount of labeled data to achieve sot-trained models, We only need to fine-tune the pre-trained results in the NLP fieldsuch as BERT [9], GPT series [10] [11] [12], XLNET [13], and erine [14]. At present, SpERT model [7], which fine-tunes BERT to encode spans, has achieved state-of-the-art performance in joint entity and relation extraction. However, the span encoding in SpERT focuses more on the deep features of the text and lacks the extraction of syntactic features such as part-of-speech, which may produce redundancies in some cases. Specifically, in the NER stage, many spans are wrongly divided into entities, which affects the performance of relation extraction. For example, figure 1 shows a sentence and the entities extracted by SpERT. Meanwhile, five spans are extracted from the sentence as entities (FP1-FP5), of which FP1, FP2, and FP3 are redundant entities. According to our observation, the part-of-speech features can help reduce redundant entities. Taking the same example in figure 1, the part-of-speech parsing result of FP1 is (State/NNP, Science/NNP, and/CC, Technolgy/NNP, Commission/NNP, Deng/NNP, Nan/NNP). Based on the part-of-speech sequence, we can easily judge that FP1 is not a correct entity because an entity rarely consists of more

than two consecutive NNP words.

Based on the above considerations, a joint extraction method based on multi-feature fusion for entities and relations is proposed. The overall model architecture follows a similar framework as SpERT. In addition to the deep features from BERT, we introduce the part-of-speech features to encode the spans in SpERT, and the proposed model is evaluated on two public data sets, CoNLL04 and ADE. Our main contributions are as follows:

- We study the impact of part-of-speech features on the word embedding of the pre-trained model. The experimental results show that part-of-speech features can effectively help text representation and improve the performance of the joint extraction model in the entity and relation extraction task.
- We explore the deep insight of part-of-speech features and find that part-of-speech features can improve the performance on entity extraction from short text. When the length of text exceeds a certain range, the performance of part-of-speech features decreases with the increase of text length.
- The proposed model achieves the state-of-the-art result of 87.19% on the NER Macro F1 in the ConLL04 data set (Previous best was 87%).

II. RELATED WORK

By sharing input features or internal hidden layer states based on shared parameters, the joint extraction model adds the loss value generated by entity recognition to the loss value generated by relation classification. Several joint extraction models based on shared parameters are introduced here. Miwa and Bansal [15] applied a syntactic analysis tree structure to extract relations. In addition, this method considers the word sequences of the sentence and pays attention to the substructure information that depends on the syntax tree. The bidirectional sequential LSTM-RNN models all two types of information, with the dependency layers stacked at the top of the sequence layer. Therefore, entity-related information can be shared during the relation extracting process. Although the proposed word sequence and dependency tree structures can extract entities and relations in a single model, the BIO tagging scheme cannot solve the entity overlap problem well.

To solve the problem of redundant entities and to ignore internal structures in the process of extracting entities and relations, Yu et al. [16] handled these problems with a new decomposition strategy, which hierarchically deconstructed the task into several sequence labeling problems. Two interrelated sub-tasks were considered: the HE extraction and the other is the TER extraction. The HE extraction task aims to distinguish all candidate head-entities that may involve the target relations, and then the TER extraction task is to identify the corresponding tail entities and relations to each extracted head-entity. Furthermore, a hierarchical boundary tagger and a multi-span decoding algorithm were applied to solve the sequence labeling problem.

Different relational triplets can overlap in sentences, which significantly increases the complexity of the relations in a sentence. Zeng et al. [17] divided all the sentences into three types based on the triplet overlap degree: Normal, EntityPairOverlap and SingleEntiyOverlap. Nowadays, most of the proposed methods so far can solve Normal-type sentences well. Zeng et al. proposed an end-to-end model with an Encoder-Decoder mechanism to obtain possible relations in the sequence for the other two types. Besides, the copy mechanism was also implemented to extract potential head and tail entities from the input sequence simultaneously.

Eberts and Ulges [7] adopted the transformer pre-training with the span method as the primary approach in joint extraction of the entities and relations. At the first step, the method extracted the candidate entities from the span. After filtering non-entities in the candidate entities, the remaining entities were merged into entity pairs, and the relations between entity pairs were extracted. Compared with the approach introduced by Eberts and Ulges [7], BERT was incorporated with the multi-head selection framework proposed by Huang et al. [18]. Based on the proposed model, after all the entities were extracted, the relation and the corresponding head entity were output simultaneously, eliminating the need for constructing entity pairs. The above methods introduce some frameworks for studying extraction models. These models usually adopt deep neural networks to embed and rarely consider shallow semantic features for input features.

Traditionally NER uses the BIO tagging scheme for sequence labeling, which results in tags being assigned to only one token, and therefore does not solve the entity nesting problem. However, the span-based method can obtain all possible token combinations in the sentence. Take the "Chicago plant" as an example, which the BIO tagging scheme will classify as one entity, namely "Chicago plant". In contrast, the span-based method generates three candidate entities: "Chicago", "plant", and "Chicago plant", each called a span. For a sequence of length T, the sum of the number of spans of different lengths is $\frac{T\times(T-1)}{2}$. Therefore, the problem of overlapping entities can be solved.

Some representative work about the span-based method is described below. In Luan's work [19], the referential resolution problem was added to the entity and relation extraction tasks. The model combined three tasks of entity recognition, relation recognition, and coreference clustering in literature, and all three tasks shared the span vector. Luan et al. [19] proposed a dynamic span graph that updated the span representation by iterative learning of edge relations (relation type and coreferential relation). In addition, the dynamic graph framework can propagate global contextual expressions, making global coding possible. Subsequently, David et al. [20] proposed the GYDIE++ model based on Luan's work [21]. Event extraction was also added to the joint extraction task, and the BiLSTM network was replaced with a BERT pre-trained model when the span vector was constructed. The span-based method can solve the problem of entity nesting, but the problem of entity redundancy [8] remained.

In the existing joint extraction models, the effect of using BERT to construct a span vector is relatively ideal, but few studies combine BERT's word embedding features with the part-of-speech features. Fatema et al. [22] studied the extraction of BERT embedded features with traditional NLP features in the extraction of relations. However, to the best of our knowledge, the effectiveness of combining part-of-speech features with BERT context embedding has not been studied for the joint extraction model of entities and relations. Therefore, our proposed model can fill this gap in this research field.

III. METHODOLOGY

In this section, the input multi-dimensional features and the architectural design of the model are introduced here.

A. Multi-dimensional features

Span Embedding: Enter a sentence, $S = w_1 w_2 \dots w_n$. We use Spacy tool [23] to split the sentence into token sequences $[t_1, t_2, \dots, t_n]$. The span-based method is applied to combine tokens of any length to generate a span sequence. The fine-tuned BERT vocabulary is adopted to encode span sequence, and then the hidden state output from the last layer of BERT is used as span Embedding.

Width Embedding: We obtain the number of tokens K contained in each span, query the word embedding table, and convert the span width to a vector with the specified dimension [24]. These embeddings are learned through back-propagation and allow the model to incorporate priors across the span [7]. In addition, adding the span width feature can filter out span entities that are too long.

CLS: CLS is the marker symbol in the BERT model. In the output from the last layer of BERT, CLS covers the semantic representation of the entire sentence [25]. Therefore, CLS features are often added to downstream classification tasks.

Pos Embedding: To encode part-of-speech features, we use NLTK [26] to convert tokens into part-of-speech symbols. Then, binary encoding is applied to design the part-of-speech vectors. The part-of-speech type in the NLTK package is fixed, and the maximum part-of-speech dimension is designed according to the total number of part-of-speech to construct the part-of-speech index table. By querying the index table, the part-of-speech embedding of each token is obtained. When binary encoding is used, and multiple tokens in the span, information about relative positions between part-of-speech can be retained.

B. Model architecture

The architecture diagram of the model is shown in Figure 2. The model input is a sentence converted to a token, and the BERT model encodes the token sequence after fine-tuning. The span method is used to combine the tokens and feed them into the span classification, and the classifier marks entity-tags for each span. The span marked None is then filtered out from the candidate entities. After pairing the remaining candidate entities, they are input into the relation classification and the context between the entity pairs. The final output is the relation tag between the entity pairs.

span classification:

The internal architecture of this stage is shown in the left half of Figure 2. The input is a span composed of tokens, and the output is the entity tag of the span. The number of entity tags is k+1 (the predefined k entity types plus one none type). The span vector is represented by $S = (t_1, t_2, \dots, t_n)$. This part is transformed into $f(t_1, t_2, \dots, t_n)$ after max-pooling function. In [7], experiments verify that max-pooling is more effective than other pooling operations in terms of text feature representation. Therefore, we also use the max-pooling method to extract the maximum feature of the span vector. The span before BERT encoding is the original text, represented by $W = (w_1, w_2, \dots, w_n)$. After each token in span is converted into part-of-speech, the part-of-speech vector $P(w_1, w_2, \dots, w_n)$ can be obtained by binary encoding. To filter entities whose span length is too long, we add the width feature of the span. The length of the input span is k, and the embedding width is E_k after querying the word embedding table. Therefore, the final input in the span classifier is:

$$x^s = f(t_1, t_2, \dots, t_n) * p(w_1, w_2, \dots, w_n) * E_k * c \quad (1)$$

where c denotes the CLS classification mark, * represents the vector splicing. The spliced vectors are passed through the softmax classifier, and each span is labeled with an entity tag (including the none tag).

relation classification: The internal architecture of this stage is shown in the right half of Figure 2. After filtering the spans classified as none, there are N span entities left. The generated candidate entity pairs have N*N groups. The model's input is N*N entity pairs, and the output is the relation tag of each entity pair. The characteristics of each entity pair input to the relational classifier are composed of three categories:

1) Let (e_1, e_2) denote the input entity pair vector, and $c(e_1, e_2)$ denote the text encoding between the entity pairs. [7] According to experiment results, it is proved that the text between entities is compared with the whole sentence and the CLS tag. From the point of view of the relation classification effect, the text between entity pairs performs best. Therefore, we also use the text between entity pairs as the relation classification feature. The entity pair and the text pass through the max-pooling layer respectively, and the generated vectors are represented as $f(e_1, e_2)$ and $f(c(e_1, e_2))$.
2) Considering the importance of the part-of-speech order at the junction of the header and tail entities with text, both the entities and the text are transformed into part-of-speech features. The entity pair before BERT encoding is denoted as (w_{e1}, w_{e2}), and the text between entity pairs is represented as $c(w_{e1}, w_{e2})$. After being converted into a part-of-speech sequence, it is encoded in binary. Entity pair and text correspond to $p(w_{e1}, w_{e2})$ and $p(c(w_{e1}, w_{e2}))$ respectively. With the inclusion of part-of-speech features, the model can filter out entity

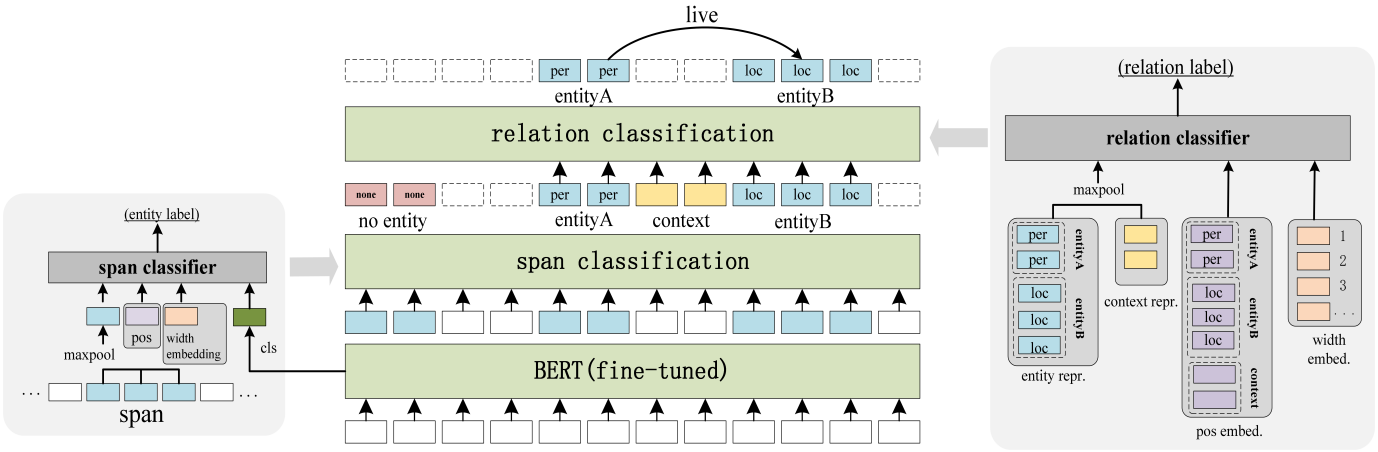

Fig. 2. Model architecture. We enter the token sequence into BERT. The output token embeddings are combined to form a span (blue). After the span classification layer, the span tag is output, and the none tag (red) is discarded. Enter the remaining entities into the relation classification after grouping them. The text between entity pairs is the context (yellow).

pairs that do not conform to the regular part-of-speech order in relation extraction.

3) The width embedding of entity pairs is also used as a feature of relation classification. The length of the token of the input entity pair is expressed as (k_1, k_2), and the converted width embedding is expressed as (w_{k1}, w_{k2}).

These three types of feature vectors are spliced and input into the relational classifier. The spliced vector is expressed as follows:

$$x^r = f(e_1, e_2) * f(c(e_1, e_2)) * p(w_{e1}, w_{e2}) * p(c(w_{e1}, w_{e2})) * (E_{k1}, E_{k2}) \quad (2)$$

where $*$ denotes vector splicing, and the spliced vector is input into the single-layer classifier and activated by the sigmoid function. The relation tag with the highest score in the sigmoid layer is used to relate the entity pairs. At the same time, the threshold h is set, and the sigmoid function can only be activated if it is greater than the threshold. Otherwise, there is no relation between the entities.

category	Train set	Test set
Loc	1541	427
Peop	1370	321
Org	786	198
Other	573	133

Table I: Part-of-speech is included in the negative samples of CoNLL04 dataset.

IV. EXPERIMENTS

A. Data set

Our model is evaluated on two public data sets.

- **CoNLL04:** The entity and relation annotations in the CoNLL04 data set [30] all derived from news reports. There are four types of entities: Location, Organization, People, and Other. There are five types of relations: Work-for, Kill, Organization-Based-In, Live-In, and Located-In. Our training set and test set are the same as [31], with 1153 sentences in the training set, and 288 sentences in the test set. The distribution of entities and relations is shown in Table I, Table II.

category	Train set	Test set
Located-In	312	94
Work-for	325	76
Kill	221	47
Live-In	421	100
Organization-Based-In	347	105

Table II: Part-of-speech is included in the negative samples of CoNLL04 dataset.

- **ADE:** The ADE data set [32] describes the impact of drug use in medical reports. The entities are categorized as Adverse-Effect and Drug, and the relation is only categorized as Adverse-Effect. Consistent with the work [7], we adopted 10-fold cross validation.

B. Baseline

On the two public data sets, we use various complex neural arrest models as the baselines.

- Relation-Metric with AT [28]: This method proposes a new neural architecture using table structures to aggregate local dependencies and metric-based features. It improves the most advanced technology without using global optimization.
- Multi-head + AT [27]: This method proposes the use of adversarial training for the joint task of entity recognition and relation extraction, and improves the current state-of-the-art baseline model.
- Multi-head [6]: This method proposes a joint neural model that does not require any manually extracted

Dataset	Model	RE Macro F1	RE Micro F1	NER Macro F1	NER Micro F1
CoNLL04	multi-head+AT [27]	61.95		83.6	
	multi-head [6]	62.04		83.9	
	Relation-Metric with AT [28]	62.29		84.15	
	Biaffine attention [29]	64.4		86.2	
	SpERT [7]	72.87	71.47	86.25	88.94
	Ours	**73.18**	**71.91**	**87.19**	**89.65**
ADE	multi-head [6]	74.58		86.40	
	multi-head + AT [27]	75.52		86.73	
	Relation-Metric [28]	77.19		87.02	
	SpERT [7](with overlap)	78.84		89.28	
	SpERT [7](without overlap)	79.24		89.25	
	Ours	**79.84**		**89.65**	

Table III: Results on CoNLL04 and ADE test sets.

Effect of Part-of-speech（1）

Without pos	Either luck or good fortune was on my side," the shaken teacher, [Donald Miller], NNP NNP said through [United Federation of Teachers] spokesman People[[Bert Shanas] .] NNP NNP IN NNP NNP NNP .
With pos	Either luck or good fortune was on my side," the shaken teacher, [Donald Miller], said through [United Federation of Teachers] spokesman People [Bert Shanas] .

Effect of Part-of-speech（2）

Without pos	Vice of the $[_1[_2$[State Science and Technology $[_3$Commission] [Deng] $_2$Nan]]$_3]_1$ NNP NNP CC NNP NNP NNP NNP called for the final solution of environmental problems.
With pos	Vice of the [State Science and Technology Commission] [Deng Nan] called for the final solution of environmental problems.

Fig. 3. Examples of entity extraction with the addition of part-of-speech features. The labels below the words are its corresponding part-of-speech symbols. "." does not have part-of-speech, therefore its corresponding part-of-speech feature is still ".". The text in brackets with the same subscript and the same color is an entity. red[*] = False Positive,green[*] =True Positive.

features or the use of any external tools. In addition, the model applies the Conditional Random Field (CRF) layers and relation extraction tasks to model the entity recognition task as a multi-head selection problem.

- Biaffine attention [29]: This method proposes a neural network model, which extends the entity recognition model based on BiLSTM-CRF to extract named entities and their relations jointly.
- SpERT [7]: This method is based on the span method, takes BERT as the core model, combines multi-dimensional features to construct a joint extraction model, which achieves state-of-the-art results on multiple public data sets.

C. Experimental Settings

In this section, the implementation details of our proposed model are covered. The pre-trained model used in the experiment is the $BERT_{BASE}$ (cased) model trained on the English corpus. The learning rate of the model is 5e-5, and the dropout is 0.1. The maximum span length is set to 10, and the dimension of width embedding is set to 25. The setting of epoch is 30, the batch size is 2, with the threshold of relational filtering is 0.4. In addition, due to the span method, the number of negative samples obtained far exceeds the number of positive samples. Therefore the maximum number of negative samples for entities and relations is set to 100. In the experiment, the dimension of each part-of-speech is set to 6. Since the maximum length of spans is set to 10, the part-of-speech encoding dimension of the spans is fixed to 60, and the additional columns of spans with length less than 10 are filled with 0. In the model training stage, the loss value is the sum of the entity recognition loss value and the relation classification loss value. Cross-entropy is used to calculate the loss value of entity recognition, and the binary cross-entropy

loss is implemented to calculate the loss value for the relation classification.

Five experiments were conducted on the CoNLL04 data set, with the same training set and test set each time. In addition, the parameter setting also remains the same in each experiment. Furthermore, the average result of the five experiments is used as the final experimental result. On the ADE data set, the 10-fold cross-validation method is implemented. In each experiment, the training set and the test set are divided in a ratio of 9:1, and the parameter settings remain unchanged. The average of each experimental result is calculated as the final result of the model. In addition, we use micro F1 and macro F1 values to evaluate the performance of the model.

D. Performance Evaluation

In this section, (i) the performance of the proposed model and (ii) the effectiveness of part-of-speech feature are evaluated.

1) **Performance of The Model**: The performance of our proposed model are evaluated on the two public data sets, CoNLL04 and ADE. The experimental results are shown in Table III.

On the CoNLL04 data set, our proposed model achieves better performance than the SpERT [7] in entity recognition and relation extraction tasks. In the entity recognition task, compared with the SpERT model [7], the Macro F1 value of our model increases by nearly 1%. In addition, our proposed model ranks first in the benchmark with the Marco F1 of 87.19% (the previous best benchmark was 87%). The excellent experimental result proves that the part-of-speech features can improve the model's performance in the entity recognition task. Since the context between the entity pairs is too long to capture in the relation extraction, the improvement of the NER task is more evident than the baseline compared with the relation extraction task. Based on the part-of-speech features, the characteristics of entities can be captured better, thus helping to achieve better performance of entity recognition.

On the ADE data set, our proposed model achieves 0.4% higher than the SpERT model [7] in the NER task with the Marco F1 of 89.65%. In addition, an increase from 79.24% to 79.84% is also achieved in the relation extraction task. Based on these experimental results, it can be found that the improvement of relation extraction compared with the baseline is more evident than the entity recognition. Unlike the CoNLL04 data set, the ADE data set applies to the medical domain, covering many medical and technical terms. Therefore, the effect of adding part-of-speech features on entity recognition is not as significant as that on the CoNLL04 data set. According to the statistics, the average sentence length of this data set is 21 words less than the average sentence length of CoNLL04 (29 words), so it is easier to capture the part-of-speech pattern in the context between the entity pairs. The order of the part-of-speech at the junction between entities and context can also help the model better identify the relation between the entity pairs.

2) **Effectiveness of part-of-speech feature**: Adding part-of-speech feature can effectively solve the redundancy problem of these two types of entities.

The first category is the entities that contain punctuation marks, as shown in the example (1) in Figure 3. The extraction model without part-of-speech feature follows the method proposed by Eberts and Ulges [7]. Since the span method is used, the extracted entities can overlap. The entity in the red brackets contains the correct entity and the full stop punctuation, which is a typical entity redundancy problem. After adding the part-of-speech feature, punctuation cannot be converted into the part-of-speech, and the converted part-of-speech symbol is still the punctuation itself. Therefore, the inclusion of part-of-speech features can eliminate redundant entities containing punctuation marks.

The second category contains unusual part-of-speech combination entities. In example (2) from Figure 3, the model without part-of-speech features extracts multiple words with the same part-of-speech connection as entities. Most models that do not contain part-of-speech features solve such problems by learning more entity features and changing the model framework. After adding the part-of-speech features, the generated classifier can filter the entities containing unusual part-of-speech combinations by learning the part-of-speech combination patterns in the entities.

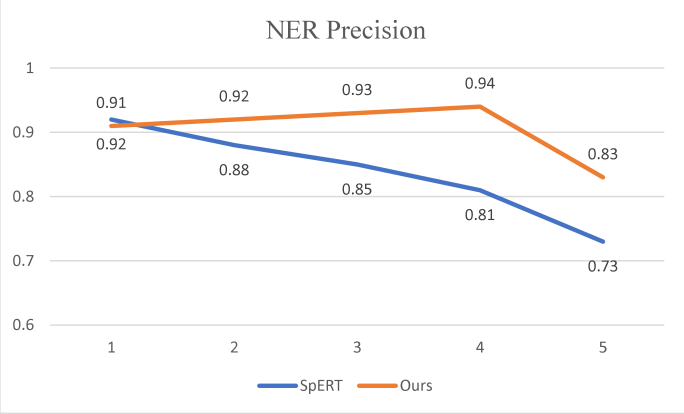

Fig. 4. The figure shows the comparison between precision values under the different entity lengths. The x-coordinate represents the length of the entity, and the y-coordinate denotes the precision value.

In addition, more experiments are conducted to explore the model in depth. On the CoNLL04 data set, the model's precision value variation is calculated based on the length of the entity in entity recognition. The SpERT [7] model is applied as the baseline, and the comparison results are shown in Figure 4. It can be seen from the figure that when the entity length is set to 1, the precision value of our model is slightly lower than the baseline. However, as the entity's length gradually increases, the fluctuation range of the model's precision value can be smaller than the baseline. The precision value of our model gradually increases first and then decreases after reaching the physical length boundary. The precision value of the baseline model has been decreasing

as the length of the entity increases. Based on this finding, it can be concluded that the span-based model can improve the recognition accuracy of short text entities after adding the part-of-speech features. Besides, when the entity length increases, the dependence of each part-of-speech feature decreases, so the recognition performance of the model decreases.

V. CONCLUTIONS

To solve entity redundancy in the span-based method, a joint entity and relation extraction model based on the multi-dimensional features is proposed. One of the contributions of this paper is applying the part-of-speech features in the joint extraction model, and the experimental results have demonstrated the effectiveness of the part-of-speech features. Compared with the baseline SpERT, our proposed model adds nearly 1% in the NER task after adding the part-of-speech features on the CoNLL04 data set. Meanwhile, the experimental results have demonstrated that the part-of-speech features can play a significant role in the joint extraction under short texts. However, it is still challenging to improve the efficiency of part-of-speech feature extraction for long text in the joint extraction model. In future work, we will validate our approach on larger public data sets and introduce more semantic features to help the model extract information under long text.

REFERENCES

[1] S. Zheng, Y. Hao, D. Lu, H. Bao, J. Xu, H. Hao, and B. Xu, "Joint entity and relation extraction based on a hybrid neural network,"Neurocomputing, vol. 257, pp. 59–66, 2017.

[2] S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou, and B. Xu, "Joint extraction of entities and relations based on a novel tagging scheme,"arXiv preprint arXiv:1706.05075, 2017.

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

[4] B. Rink and S. Harabagiu, "Utd: Classifying semantic relations by combining lexical and semantic resources," inProceedings of the 5th international workshop on semantic evaluation, 2010, pp. 256–259.

[5] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," inProceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009), 2009, pp. 147–155.

[6] G. Bekoulis, J. Deleu, T. Demeester, and C. Develder, "Joint entity recognition and relation extraction as a multi-head selection problem,"Expert Systems with Applications, vol. 114, pp. 34–45, 2018.

[7] M. Eberts and A. Ulges, "Span-based joint entity and relation extraction with transformer pre-training,"arXiv preprint arXiv:1909.07755, 2019.

[8] K. Dixit and Y. Al-Onaizan, "Span-level model for relation extraction," inProceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 5308–5314.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding,"arXiv preprint arXiv:1810.04805, 2018.

[10] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[11] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskeveret al., "Language models are unsupervised multitask learners,"OpenAI blog, vol. 1, no. 8, p. 9, 2019.

[12] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askellet al., "Language models are few-shot learners,"arXiv preprint arXiv:2005.14165, 2020.

[13] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding,"Advances in neural information processing systems, vol. 32, 2019.

[14] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "Ernie: Enhanced language representation with informative entities,"arXiv preprint arXiv:1905.07129, 2019.

[15] M. Miwa and M. Bansal, "End-to-end relation extraction using lstms on sequences and tree structures,"arXiv preprint arXiv:1601.00770, 2016.

[16] B. Yu, Z. Zhang, X. Shu, Y. Wang, T. Liu, B. Wang, and S. Li, "Joint extraction of entities and relations based on a novel decomposition strategy,"arXiv preprint arXiv:1909.04273, 2019.

[17] X. Zeng, D. Zeng, S. He, K. Liu, and J. Zhao, "Extracting relational facts by an end-to-end neural model with copy mechanism," inProceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 506–514.

[18] W. Huang, X. Cheng, T. Wang, and W. Chu, "Bert-based multi-head selection for joint entity-relation extraction," inCCF International Conference on Natural Language Processing and Chinese Computing. Springer, 2019, pp. 713–723.

[19] Y. Luan, L. He, M. Ostendorf, and H. Hajishirzi, "Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction,"arXiv preprint arXiv:1808.09602, 2018.

[20] D. Wadden, U. Wennberg, Y. Luan, and H. Hajishirzi, "Entity, relation, and event extraction with contextualized span representations,"arXiv preprint arXiv:1909.03546, 2019.

[21] Y. Luan, D. Wadden, L. He, A. Shah, M. Ostendorf, and H. Hajishirzi, "A general framework for information extraction using dynamic span graphs,"arXiv preprint arXiv:1904.03296, 2019.

[22] F. Hasan, A. Roy, and S. Pan, "Integrating text embedding with traditional nlp features for clinical relation extraction," in2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI). IEEE, 2020, pp. 418–425.

[23] M. Neumann, D. King, I. Beltagy, and W. Ammar, "Scispacy: Fast and robust models for biomedical natural language processing,"arXiv preprint arXiv:1902.07669, 2019.

[24] K. Lee, L. He, M. Lewis, and L. Zettlemoyer, "End-to-end neural coreference resolution,"arXiv preprint arXiv:1707.07045, 2017.

[25] A. Rogers, O. Kovaleva, and A. Rumshisky, "A primer in bertology: What we know about how bert works,"Transactions of the Association for Computational Linguistics, vol. 8, pp. 842–866, 2020.

[26] E. Loper and S. Bird, "Nltk: The natural language toolkit,"arXiv preprint cs/0205028, 2002.

[27] G. Bekoulis, J. Deleu, T. Demeester, and C. Develder, "Adversarial training for multi-context joint entity and relation extraction,"arXiv preprint arXiv:1808.06876, 2018.

[28] T. Tran and R. Kavuluru, "Neural metric learning for fast end-to-end relation extraction,"arXiv preprint arXiv:1905.07458, 2019.

[29] D. Q. Nguyen and K. Verspoor, "End-to-end neural relation extraction using deep biaffine attention," inEuropean Conference on Information Retrieval. Springer, 2019, pp. 729–738.

[30] 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, Tech. Rep., 2004.

[31] P. Gupta, H. Schütze, and B. Andrassy, "Table filling multi-task recurrent neural network for joint entity and relation extraction," inProceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 2537–2547.

[32] H. Gurulingappa, A. M. Rajput, A. Roberts, J. Fluck, M. Hofmann-Apitius, and L. Toldo, "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.