AttSC: A span-based Model for Joint Entity and Relation Extraction in Chinese Legal Text

Daiyang Luan† Haorui Li† Yuanhao Yue‡
†Chien-shiung Wu College, Southeast University ‡College of Artificial Intelligence, Southeast University
{213180230, 213180146}@seu.edu.cn

Abstract

Relation extraction is an important task in Nature Language Process. Given a predefined set of entities and a sentence, the goal is to extract triplets which contain the head entity, the tail entity and the relation between them. The task consists of two subproblems. Namely, the Named Entity Recognition and Relation Classification. While common methods tackle them separately, more recent works use joint models. Recently, Eberts and Ulges proposed SpERT[2], a simple but effective span-based model that takes BERT as backbone and use two FFNNs to classify span and relation respectively. Unlike previous works, SpERT dramatically reduces model training complexity by adopting negative sampling. However, existing models have a way of ignoring some significant features, including span-specific features and sentence-level contextual information. To address this problem, we propose AttSC, a new attention-based model to better jointly extract the triplets from raw text.

Meanwhile, because of the uniqueness of Chinese text, we introduce multiple embeddings to the existing BERT pretrained model. Experiments are conducted to demonstrate that our approach outperforms the baseline method.

1 Introduction

Extracting entities and their relations from raw text is a significant information extraction task in NLP. A common approach is to constrct pipeline systems which treat named entity recognition(NER) and relation classification(RC) as two separate tasks. However, it has been proved that end-to-end systems which jointly learn to extract entities and relations have the potential to obtain higher performance for its ability to avoid error accumulation, entity redundancy and to share information between both two tasks.

Models based on neural networks has obtained state-of-the-art performances in joint extraction of entities and relations. Miwa and Bansal[3] applied bidirectional tree-structured LSTMs to model dependency parse tree between an entity pair to predict the relation type. In their work, however, relation extraction still causes entity redundancy, which gives rise to error rate and computation complexity. Moreover, dependency parse trees can only handle sentence-level languages. Zhang et al.[4] integrated implicit syntactic information by using latent feature representations extracted from a pre-trained BiLSTM-based dependency parser. D. Q. Nguyen and K. Verspoor[1] extended a BiLSTM-CRF-based entity recognition model with a deep biaffine attention layer to model second-order interactions between latent features for relation classification, specifically attending to the role of an entity in a directional relationship.

In span-based models, any token subsequence (or span) constitutes a potential entity, and a relation can hold between any pair of spans. Markus Eberts and Adrian Ulges[2] proposed SpERT, where all candidate spans are classified for possible entities.The core of their approach is pre-trained language models. The finetuned BERT embeddings are fed into relation classifier together with the context between both entities. In comparison to deep learning models, their approach worked well in CoNLL04 dataset with the F1-score 88.94, but still no larger than 80 in SciERC(70.33).

Different tokens in spans should contribute differently to span representation. But in SpERT[2], each token in spans is treated equally important. In other words, the span-specific features, which may contribute to the performance of the model significantly, are ignored. What's more, semantic information is not adequately captured by the max-pooling function as the fusion function in the original model. Sentence-level contextual information is ignored in both processes of span classification and relation classification, which may be important compensation information for both ones. To address the problems mentioned above, the AttSC model is proposed with attention-based span-specific features.

2 Related Work

2.1 SpERT

The input sequence is fed into the byte-pair encoder(BPE) to obtain a sequence of n tokens. After that, the BPE tokens are passed through fine-tuned BERT for an embedding sequence $(e_1, e_2, ..., e_n, c)$ where c represents a special classifier token capturing the overall sentence context. The span classifier takes an arbitrary candidate span as an input and takes the width(w) and context(c) into account to tell whether this span belongs to a certain entity class.

$$x^s := f(e_i, e_{i+1}, ..., e_{i+k}) \circ w_{k+1} \circ c$$

$$\hat{y}^s = \text{softmax}\left(W^s \cdot x^s + b^s\right)$$

Here f is a fusion function and it is found that max-pooling works best. The relation between two entities is classified based on the maxpooling of BERT embeddings of both entities $e(s_1), e(s_2)$ and their context representations $c(s_1, s_2)$.

$$x_1^r := e(s_1) \circ c(s_1, s_2) \circ e(s_2)$$

$$x_2^r := e(s_2) \circ c(s_1, s_2) \circ e(s_1)$$

Using a single layer classifier, any relation with a score higher than a certain threshold is considered activated.

$$\hat{y}_{1/2}^r := \sigma\left(W^r \cdot x_{1/2}^r + b^r\right)$$

The training loss is a joint loss function for entity classification and relation classification

$$\mathcal{L} = \mathcal{L}^s + \mathcal{L}^r$$

Here $\mathcal{L}^s$ is the cross entropy loss of the span classifier and $\mathcal{L}^r$ is the binary cross entropy over relation classes.

2.2 Attention Mechanism

An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. In practice, we compute the attention function on a set of queries simultaneously, packed together into a matrix Q. The keys and values are also packed together into matrices K and V . We compute the matrix of outputs as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

The weighted sum is a selective summary of the information contained in the values, where the query determines which values to focus on. Attention is a way to obtain a fixed-size representation of an arbitrary set of values, dependent on the query. It is great because it allows our model to focus on certain parts of the source. What's more, it provides some interpretability.

3 Attention-based Span Classification

As our research is based on the SpERT[2] model, we need to reproduce this paper as our baseline. Then efforts will be made to improve the performance and to test the effect over different domains.

In original SpERT, max-pooling was chosen as the fusion function to concatenate all the embeddings in a span in feature level. Thus the most distinctive features are retained. Here attention

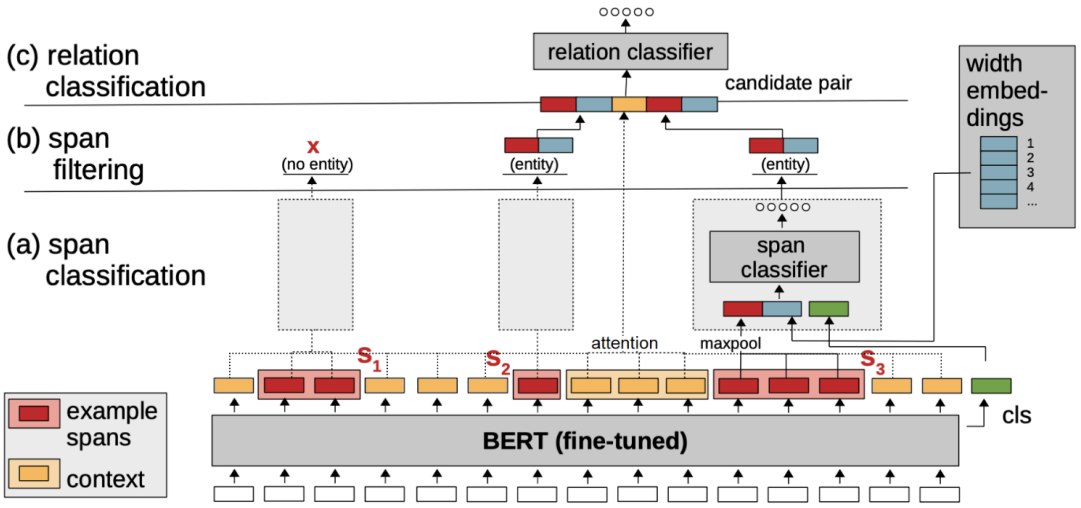

Figure 1: Our AttSC(Attention-based Span Classification) model.

can be used to concatenate embeddings in semantic level. Moreover, the relation between two entities does not necessarily appear exactly between them.

Specifically, we define a sentence and a span from raw text as:

$$\text{Sentence:} \quad \mathcal{S} = (t_1, t_2, ..., t_n)$$
$$\text{Span:} \quad s = (t_i, t_{i+1}, ..., t_{i+j})$$

where t denotes tokens and subscripts denote token indexes.

The embedding of sentences and spans are defined as:

$$\mathcal{B}_{\mathcal{S}} = [X_0, X_1, ..., X_n]$$
$$\mathcal{B}_s = [X_i, X_{i+1}, ..., X_{i+j}]$$

where X_i denotes the embedding of token t_i. Here we use multiple embedding to replace BERT embedding, as is discussed in 3.1.

3.1 Multiple Embedding

Compared with the English language, joint extraction on Chinese text faces more challenges. It is known that Asian languages like Chinese are naturally logographic. There are no conventional linguistic features (like capitalization) available. Moreover, Chinese characters and words hold complex relations. Given no delimiter between characters, a word can be comprised of a single character or several characters (i.e., n-char). Also, the meaning of a Chinese word can be inferred by its constituent characters. Furthermore, Chinese characters often have hieroglyphic forms. The same radical often indicates similar semantics. Hence, we may apply the multiple embedding to our joint extraction system.

Radical embeddings. Chinese characters are hieroglyphic in nature. The same parts (i.e., radical) in different characters often share the same meaning. The use of radicals enables us to better infer the semantics of the characters that only appear in the test set but not in the training set.

Character embeddings. Characters are elementry elements in Chinese. The meaning of a word can be inferred via its component characters because they have their own meanings. For example, the word *confidence* in Chinese is composed of Chinese characters *self* and *belief*. By using character embeddings, we can get semantic knowledge from Chinese characters.

Word embeddings. As a high-level representation, word embeddings are the most common way to exploit sementic expressions in Chinese language. If a word is not in our vocabulary, we may

initialize its embedding with its constituent characters.

3.2 Span Representation

In our AttSC model, span representation is composed of four parts, namely **a**) concatenation of span head and tail representations, **b**) span-specific representation, **c**) sentence-level contextual representation, and **d**) span width embedding.

3.2.1 Concatenation of span head and tail representations

We define concatenation result of span s as: $\mathcal{H}_s = [X_i; X_{i+j}$ Here if span s consists of only one token, then we let $X_{i+j} = X_i$.

3.2.2 Span-specific representation

We use symbol $\mathcal{F}_s$ to illustrate the span-specific representation of span s and it is calculated as follows:

$$\mathcal{V}_k = \mathbf{MLP}_k(X_k) \qquad \text{for } k \in [i, i+j]$$

$$\alpha_k = \frac{\exp(\mathcal{V}_k)}{\sum_{m=i}^{i+j} \exp(\mathcal{V}_m)}$$

$$\mathcal{F}_s = \sum_{m=i}^{i+j} \alpha_m X_m$$

Here **MLP** means the MLP attention. By using span-specific representation, we can add more weight to the tokens which are of more significance.

3.2.3 Sentence-level contextual representation

Take $\mathcal{F}_s$ as *query*, $\mathcal{B}_\mathcal{S}$ as *key* and *value* respectively, the sentence-level contextual representation for span s is defined as:

$$\mathcal{T}_s = \mathbf{Attention}(\mathcal{F}_s, \mathcal{B}_\mathcal{S}, \mathcal{B}_\mathcal{S})$$

Here information beneficial for span classification will be assigned a heavy weight, and the contextual representation will be taken to constitute span representation.

3.2.4 Span width embedding

Fixed-size embedding for span of width $1, 2, ...$ can be learned during the training process. Thus the embedding $\mathcal{W}_{j-i+1}$ can be looked up from a vector learned.

3.2.5 Span classification and filtering

The final representation for classification of span s is:

$$\mathcal{R}_s = [\mathcal{T}_s, \mathcal{F}_s, \mathcal{H}_s, \mathcal{W}_{j-i+1}]$$

The representation $\mathcal{R}_s$ is fed into a multi-layer fed-forward neural network and a softmax classifier:

$$y_s = \mathbf{SoftMax}(\mathbf{FFNN}(\mathcal{R}_s))$$

Spans which probably consists of certain entities are kept and those classified into the *NoneEntity* class are excluded to form a predicted entity set E. The relation classification is performed on $\{E \bigotimes E\}$. Hence, the scale of candidate entities is smaller and the complexity of computation is reduced.

4 Relation Classification and Filtering

Relation representation for classification composes of three parts, namely **a**) concatenation of relation tuple representations, **b**) local contextual representation, **c**) sentence-level contextual representation.

4.1 Concatenation of relation tuple representations

The representations of spans s_1, s_2 are fed into a multi-layer fed-forward neural network to reduce their dimensions and the result is:

$$\mathcal{H}_r = [\mathbf{FFNN}(\mathcal{R}_{s_1}), \mathbf{FFNN}(\mathcal{R}_{s_2})]$$

4.2 Local contextual representation

Let $\mathcal{H}_r$ be query, multiple embedding of the span between both entities $\mathcal{B}_c$ be key and value respectively, the attention-based contextual representation is calculated as:

$$\mathcal{F}_r = \mathbf{Attention}(\mathcal{H}_r, \mathcal{B}_c, \mathcal{B}_c)$$

Here $\mathcal{B}_c$ is defined as:

$$\mathcal{B}_c = (X_m, X_{m+1}, ..., X_n)$$

where $m-1$ is the index of the end token of the first entity while $n+1$ is the index of the beginning token of the second entity.

4.3 Sentence-level Contextual Representation

Let $\mathcal{H}_r$ be query, multiple embedding of the whole sentence $\mathcal{B}_S$ be key and value respectively, the attention-based contextual representation is calculated as:

$$\mathcal{T}_r = \textbf{Attention}(\mathcal{H}_r, \mathcal{B}_S, \mathcal{B}_S)$$

4.4 Relation classification

The final representation of the relation between s_1 and s_2 is

$$\mathcal{R}_r = [\mathcal{H}_r; \textbf{FFNN}_{\mathcal{F}}(\mathcal{F}_r); \textbf{FFNN}_{\mathcal{T}}(\mathcal{T}_r)]$$

Here $\mathcal{F}_r$ and $\mathcal{T}_r$ are fed into different multi-layer fed-forward neural networks before being concatenated. Then $\mathcal{R}_r$ passes through a multi-layer fed-forward neural network and a softmax classifier to determine the relation distribution:

$$y_r = \textbf{SoftMax}(\textbf{FFNN}(\mathcal{R}_r))$$

5 Domain Changing via Domain Feature Enhancement

The pre-trained language model BERT benefits many NLP tasks thanks to its abundant prior knowledge. However, it captures the general language information from the large-scale corpus during the training procedure, which leads to the lack of the task-specific and domain-specific knowledge.

There are researches aiming at integrating knowledge into the BERT model. Some incorporates explicit syntactic constraints into attention mechanism in order to guide the text modeling with syntax. Works on embedding the knowledge bases into pre-trained language models contribute to introducing domain knowledge.

The encoding of the pre-trained model tends to capture the general text representation but is short of domain knowledge. In order to make up for the lack of domain information, we add domain feature enhancement into the encoder. The accuracy of the recognized entities is crucial for the performance of triplet extraction. Hence, the lexicon feature for the entities is worthy of exploration. Towards the triplet extraction on the Chinese legal documents, we first build up a specific lexicon $Lexicon_{\text{legal}}$ as the domain feature.

We collect the specific names of all kinds of legal terms and and their common statements recording in the judgment documents. The lexicon of specific domain knowledge includes the possible expressions of all legal terms, both in written and spoken language. The encoder with domain feature enhancement adds special weight to spans which appear in our lexicon.

6 Model Training

We use symbol $\mathcal{L}^s$ to denote the cross-entropy loss of span classification and $\mathcal{L}^r$ to denote the binary cross-entropy loss of relation classification. Then the final loss function of our model is

$$\mathcal{L} = \mathcal{L}^s + \theta\mathcal{L}^r$$

Here θ is a parameter which can be modified. In practice, the performance of relation classification is usually worse than span classification. Hence, a larger weight is applied to $\mathcal{L}^r$ in order to make our model focus more on relation classification. In our experiment, we let $\theta = 1.5$.

In model training, parameter matrices of fed-forward neural networks and attention equations are learned, multi-embedding of tokens fine-tuned.

7 Experiment

In this section, we first present details of our dataset. Then results are compared by relations between our approach and the state-of-art SpERT method.

7.1 Experimental Settings

Negative sampling is adopted during model training to improve model performance and robustness.

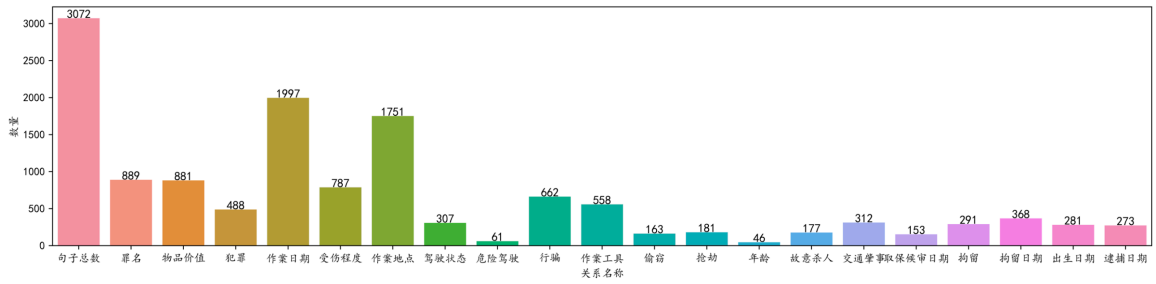

Figure 2: Distribution of sentence count and relation types

We set the negative sampling rate to 30, batch size for model training to 8, dropout to 0.2 and width embedding dimension to 50. **FFNN**$_{\mathcal{F}}$ and **FFNN**$_{\mathcal{T}}$ contain three fully connected layers; and all the other **FFNN**s contain two layers.

7.2 Dataset

7.2.1 CoNLL04 dataset

CoNLL04 composes of news articles from outlets such as WSJ and AP, We follow the training/dev/test split in (Adel and Schu tze,2017[8]; Bekoulis et al., 2018[9]), which consists 910 articles for training, 243 for development and 288 for test.

7.2.2 The Chinese legal dataset

The Chinese legal dataset for relation extraction is based on legal documents. Entities and relations among them are labeled.

There are mainly three types of documents: *verdicts*, *law enforcement records* and *criminal fact descriptions*. All samples are stored in a list and in each sample, a dictionary is used to record entity types, entity positions, relation types and positions of head and tail entity.

Code Listing 1: A sample in our new dataset

```
{"tokens": "依照《中华人民共和国刑法》第一百三十三条之一、第六十七条第三款、第七十二条第一款、第三款、第七十三条第一款、第三款、第五十二条、第五十三条之规定，判决如下：被告人程善宏犯危险驾驶罪，判处拘役二个月，宣告缓刑二个月，并处罚金一万五千元。",
"entities": [{"type": "Party", "start": 81, "end": 84},
{"type": "Criminal_Charge", "start": 85, "end": 90}], "relations": [{"type": "crime_name", "start": 0, "end": 1}], "org_id": 1001},
{"tokens": "依照《中华人民共和国刑法》第一百三十三条之一、第六十七条第三款、第七十二条第一款、第三款、第七十三条第一款、第三款、第五十二条、第五十三条之规定，判决如下：被告人钱昌国犯危险驾驶罪，判处拘役二个月，宣告缓刑二个月，并处罚金一万五千元。","entities": [{"type": "Party", "start": 81, "end": 84},
{"type": "Criminal_Charge", "start": 85, "end": 90}], "relations": [{"type": "crime_name", "start": 0, "end": 1}], "org_id": 1002},
{"tokens": "依照《中华人民共和国刑法》第一百三十三条之一、第六十七条第三款、第七十二条第一款、第三款、第七十三条第一款、第三款、第五十二条、第五十三条之规定，判决如下：被告人郭圣开犯危险驾驶罪，判处拘役二个月，宣告缓刑二个月，并处罚金一万五千元。",
"entities": [{"type": "Party", "start": 81, "end": 84},
{"type": "Criminal_Charge", "start": 85, "end": 90}], "relations": [{"type": "crime_name", "start": 0, "end": 1}], "org_id": 1003},
```

Our dataset contains 3072 sentences and 20 relation types. The distribution of them is shown in Figure 2.

7.3 Results

Table 1 and table 2 shows the performance of both models on CoNLL04 while table 3 and table 4 shows that on Chinese Legal Dataset.

Table 1: SpERT(CoNLL04)

Type	Precision(%)	Recall(%)	F1(%)
Live	72.62	77.00	76.62
OrgBI	73.47	68.57	70.94
Kill	87.23	87.23	87.23
LocIn	74.36	61.70	67.44
Work	67.95	69.74	68.83
micro	74.88	71.33	73.06
macro	75.85	72.85	74.21

Table 2: AttSC(CoNLL04)

Type	Precision(%)	Recall(%)	F1(%)
Live	80.18	83.51	81.82
OrgBI	79.11	73.87	76.40
Kill	90.92	88.83	89.86
LocIn	74.60	66.14	70.12
Work	68.23	73.54	70.79
micro	79.05	74.99	76.97
macro	79.86	74.68	77.18

Table 3: SpERT(CLD)

Type	Precision(%)	Recall(%)	F1(%)
Imprisonment	74.60	84.62	79.29
Crime	64.41	69.23	66.73
Probation	71.43	72.73	72.07
Micro	70.15	75.53	72.70
Macro	70.08	75.68	72.73

Table 4: AttSC(CLD)

Type	Precision(%)	Recall(%)	F1(%)
Imprisonment	80.91	86.13	83.44
Crime	72.52	70.63	71.56
Probation	78.21	80.15	79.17
Micro	77.13	75.88	76.50
Macro	78.56	76.93	77.74

8 Conclusion

We have presented AttSC, a span-based model using BERT pretrained model as its core and featuring multi-embeddings in Chinese language. It outperforms the current SpERT model by introducing the attention mechanism to span classification so that more semantic features can be obtained.

In the future, we plan to investigate more elaborate forms of context for relation classifiers. Employing additional syntactic features or learned context, while maintaining an efficient exhaustive search, appears to be a promising challenge.

References

[1] Nguyen D Q, Verspoor K. End-to-end neural relation extraction using deep biaffine attention[C]//European Conference on Information Retrieval. Springer, Cham, 2019: 729-738.

[2] Eberts M, Ulges A. Span-based joint entity and relation extraction with transformer pretraining[J]. arXiv preprint arXiv:1909.07755, 2019.

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

[4] Zhang M, Zhang Y, Fu G. End-to-end neural relation extraction with global optimization[C]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017: 1730-1740.

[5] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[6] Xu C, Wang F, Han J, et al. Exploiting multiple embeddings for chinese named entity recognition[C]//Proceedings of the 28th ACM international conference on information and knowledge management. 2019: 2269-2272.

[7] Chen Y, Sun Y, Yang Z, et al. Joint Entity and Relation Extraction for Legal Documents with Legal Feature Enhancement[C]//Proceedings of the 28th International Conference on Computational Linguistics. 2020: 1561-1571.

[8] Adel H, Schütze H. Global normalization of convolutional neural networks for joint entity and relation classification[J]. arXiv preprint arXiv:1707.07719, 2017.

[9] Bekoulis G, Deleu J, Demeester T, et al. Joint entity recognition and relation extraction as a multi-head selection problem[J]. Expert Systems with Applications, 2018, 114: 34-45.