

# CLASSIFICATION ATTENTION FOR CHINESE NER

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

The character-based model, such as BERT, has achieved remarkable success in Chinese named entity recognition (NER). However, such model would likely miss the overall information of the entity words. In this paper, we propose to combine priori entity information with BERT. Instead of relying on additional lexicons or pre-trained word embeddings, our model has generated entity classification embeddings directly on the pre-trained BERT, having the merit of increasing model practicability and avoiding OOV problem. Experiments show that our model has achieved state-of-the-art results on 3 Chinese NER datasets.

## 1 INTRODUCTION

The study of NER is of great importance as a pre-processing step for a variety of downstream applications in Information Extraction. Through supervised learning, the NER task can be cast to a sequence labeling task which involves identifying both entity boundaries and entity types Li et al. (2018). By doing so, a named entity is considered correctly recognized only if both of its boundary and type match the ground truth. Sang & De Meulder (2003) Pradhan et al. (2012)

Due to the differences in language structure, Chinese sentences are constructed in units of character with no uppercase identifier to special words such as names or locations, making it inherently unfriendly to the boundary identifying problem. To apply the word-based model in Chinese, traditionally, a segmentation step is required ahead of time in a pipeline architecture, where the errors of word boundaries could pass to the consequent NER process.

In recent years, character based Chinese NER models including BILSTM-CRF Lample et al. (2016) and BERT Devlin et al. (2018) have become dominant and achieved state-of-the-art results. However, using only character-based model would lead to potential loss of overall information. A common mitigation is to add external lexicon as a reference. Whether directly integrated into character-based models by means of word embeddings, or directly repairing the boundaries of entities recognized by character-based models, the performance of lexicon-assisted model is strongly related to the quality of the lexicon. However, building a complete pre-defined lexicon is costly and can hardly handle OOV issues.

In this paper, an entity classification-assisted model is proposed to alleviate this problem. Above all, we design a mechanism to generate embeddings for each entity class using only the character-based BERT pre-trained model. It mainly consists of two procedures. The first step is to form word embeddings of entities appearing in the training sentences through character embeddings and the second step is to aggregate entity embeddings by category and generate classification embeddings. Benefited from the inherent contextual information given in BERT representation, entities appearing under identical context can have identical representations. To further eliminate noise and extract common attributes between entities in the same class, a weighted projection removal method is applied additionally.

After that, we designed a novel Attention mechanism to integrate entity classification embeddings into the character-based BERT model. The input sources of our attention architecture consists of a set of preprocessed entity classes and an input sentence. In order to get a series of weighted representation of the input sentence, the traditional Scaled Dot-Product Attention is revised and increased in dimension, providing greater weight to characters identical to each entity class.

Results show that our model outperforms other Chinese NER models over a variety of datasets across different domains without external resources. Our code will be released at XXXX

## 2 RELATED MODELS

Although character sequence labeling has been the dominant approach for Chinese NER, many works seek to add further information into the model.

As mentioned above, word-based NER models suffer from the segmentation-NER pipeline, one of the solutions is to import a multitask structure. It is a common way to share mutual information between related tasks and boost the general performance. Peng & Dredze (2016) jointly trained CWS (Chinese word segmentation) task with NER task and yield significant improvements in NER for Chinese social media. To further explore language regularities, more works Yang et al. (2016) Ruder12 et al. (2017) consider adding additional related tasks into the mission, employing parameter sharing or selective sharing. Transfer learning is another popular way of improvement. Sun et al. (2019a) Feng et al. (2018) utilized the abundant resources in English to boost the NER performances in low resource languages including Chinese. Besides, Zhang & Yang (2018) investigate a lattice-structured LSTM model for Chinese NER, which encodes a sequence of input characters as well as all potential words that match a lexicon, having the merit of leveraging explicit word information over character sequence labeling without suffering from segmentation error. Liu et al. (2019) further adjust the structure of the lattice model, assigning word information to a single character and ensuring that there is no shortcut paths between characters to prevent the model from degenerating into a partial word-based model. Different from the previous, our method does not require external resources including labeled data for other tasks or predefined lexicons, making the model more practical.

Recently, many works exploit language processing with deep language models to improve the performance of downstream NLP tasks, such as ELMo Peters et al. (2018), GPT Radford et al. (2018), BERT Devlin et al. (2018), ERNIE Sun et al. (2019b) Zhang et al. (2019), XLNET Yang et al. (2019). These methods first pretrain neural networks on large-scale unlabeled text corpora, and then finetune the models or representations on downstream tasks. The dynamic embeddings learned from these models outperform the traditional static embeddings by containing more contextual information as well as relieving the polysemy problem. Tagging Models based on these structures bring impressive results Sun et al. (2019a) Kaneko & Komachi (2019) Lee et al. (2019). Our method proposed a way to extract the representation of the entity classification embeddings through the rich information contained in the pre-trained language model, which eliminates the need for external word embeddings.

What's more, the attention mechanism has shown impressive performance on many NLP tasks. Kaneko & Komachi (2019) investigate the effect of utilizing information not only from the final layer but also from intermediate layers of a pre-trained language representation model to detect grammatical errors. Rei et al. (2016) proposed a novel architecture for combining the character-based representation with the word embedding by using an attention mechanism, allowing the model to choose which information to use from each information source dynamically. Zhu et al. (2019) investigate a practical Convolutional Attention Network for Chinese NER which not depending on any external resources. The Scaled Dot-Product Attention proposed by Vaswani et al. (2017) is one of the mainstream models currently Kaneko & Komachi (2019), Devlin et al. (2018), Sun et al. (2019b), Zhang et al. (2019), Yang et al. (2019), Lee et al. (2019). In this paper, we revise the Scaled Dot-Product Attention to Classification Attention which would give a weighted representation of the input sentences through a series of entity classes.

## 3 MODEL

### 3.1 OVERALL ARCHITECTURE

As shown in Figure 1, the overall architecture of the proposed model is mainly composed of three parts: Embedding extraction for entity class and Classification Attention. The embedding extraction step is performed firstly in a pipeline structure and the results will be aggregated to the attention model. Details of the 2 parts will be elaborated as follow.

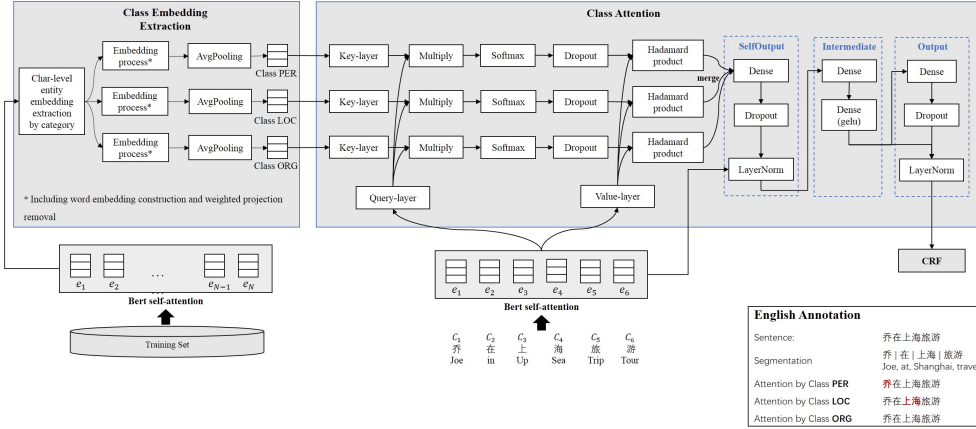


Figure 1: The overall architecture of the proposed model. The detailed structure for Classification Embedding Extraction is will be shown in following Section

### 3.2 EMBEDDING EXTRACTION FOR ENTITY CLASS

Word Lexicon has played an important role in dealing model NER problems. However, a high-quality lexicon construction requires a lot of manpower and might be incompetent to handle OOV issues. Since the ideal word embedding assumes that words with identical classes would have clustered representations, it is suggested that the embedding of known entities may help to find new entities of the same class.

Pre-trained with a large amount of Chinese corpus, the last hidden layer of BERT integrates context information into each character of the input sentences. It not only excludes synonym interference compared with traditional static embeddings, but also provides potential information for the embedding vector of entities in the context. Inspired from Arora et al. (2016), in this paper, we illustrate a simple way to extract word embeddings of entities in the hidden outputs of BERT. Figure 2.a gives the flow chart of the whole procedure.

Let  $X = \{x_1, x_2 \dots x_L\}$  denotes the input sentence in character level, where  $L$  is the total length and  $Y = \{y_1, y_2 \dots y_L\}$  denotes the label sequences accordingly. Let  $H = \{h_1, h_2 \dots h_L\}$  be the last hidden layer of BERT. Under the BMES label scheme Wang & Xu (2017), both the class and location of each entity can be inferred by the label sequence  $Y$ . To fix the size, the embedding of an entity is the averaged embedding of its composed character.

$$\begin{aligned}
 \text{Emb}(\text{Word}) &= \text{Emb}(\text{char}_m, \text{char}_{m+1}, \dots, \text{char}_n) \\
 &= \frac{1}{n-m} \sum_{i=m}^n \text{Emb}(\text{char}_i) \\
 &= \frac{1}{n-m} \sum_{i=m}^n h_i
 \end{aligned}$$

Different from the original method in Arora et al. (2016), the smooth inverse frequency is abandoned due to the dynamic semantic information encoded in  $H$ . To further get the most representative embedding of a given entity, the weighted projection of the word embeddings on their first singular vector is removed. Let  $Matrix$  be the collected embeddings of a certain class, whose columns are  $\{Emb_1, Emb_2, \dots, Emb_S\}$ , and let  $u$  be its first singular vector and  $\alpha$  be the weight parameter  $\alpha \in [0, 1]$ . For all  $Emb_i$ , do:

$$Emb_i = Emb_i - \alpha u u^T Emb_i$$

It is assumed that when the gap between the training data and the testing data is large,  $\alpha$  may takes a larger value to show a more common representation of the entity class. Finally, the embedding of a certain entity class can be gathered through the output of averagePooling layer. The whole procedure is shown in Figure2(a).

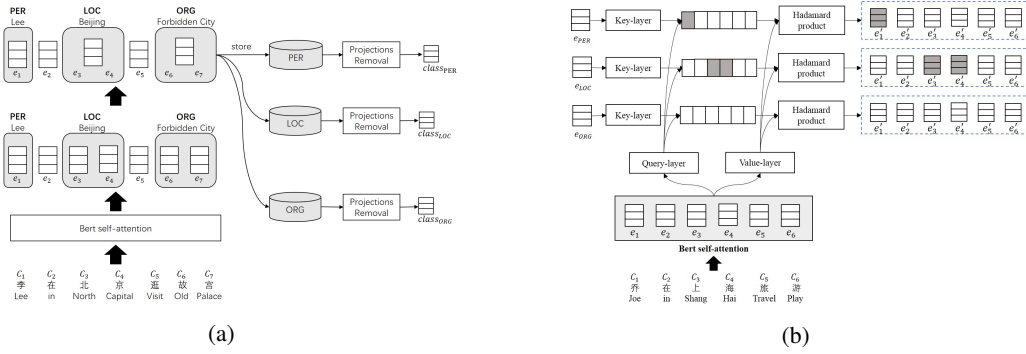


Figure 2: Figure2(a) :Embedding extraction for entity classification. Figure 2(b) Illustration of the attention architecture.

### 3.3 CLASSIFICATION ATTENTION

As shown in Figure 1, the Classification Attention architecture is designed on the basis of Multi-Head Attention proposed by Transformer Vaswani et al. (2017). The biggest difference for this architecture lies in the input source which contains a set of entity classification embeddings and character embeddings of the input sentence. It aims to get a series of weighted representation of the input sentences so that characters similar to a particular entity class could have greater feature weights.

Firstly, consider a situation with only one entity class. Matrix  $Q$  and Matrix  $K$  are adopted to extract the characteristics of the input sentence and the entity. Here, the length of the input entity class can be seen as an input sentence with only one character. By applying the traditional Scaled Dot-Product Attention, the similarity between each character of the input sentence and the entity class can be calculated by:

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

The output product gives indication of how likely a character in the input sentence is to be the input entity. Since the final result we would like to get is the weighted embedding of the input sentence. So, the  $V$  matrix in our architecture is calculated from the input sentence other than the entity. The final result can be got through

$$\text{Attention}(Q, K, V) = \text{similarity}(Q, K) \odot V$$

Figure 2(b) gives an illustration of the attention architecture for multiple entity classes. For the sake of simplicity, the softmax step and the dropout step are omitted, and the normalized scaling dot product between the entity classification embedding and the sentence embedding after the Key-layer and then Query-layer respectively is directly displayed. The intensity of the weight is proportional to the depth of color giving indication of how likely the input character is to the particular entity. After applying the Hadamard product, the weighted representation of the original sentence after Value-layer is finally obtained.

During the calculation, Multi-head mechanism is adopted for each entity class and the entire calculation is done once by a matrix one-dimensional higher than BERT. After the attention stage, the merged output is then passed along with the original character embeddings of the input sentence through the same follow-up layers as BERT, namely Self-output layer, Intermediate layer, and Output layer. Figure 1 gives the detailed structure of the 3 layers where two residential nets are applied to ensure the validity of the deep network.

## 4 EXPERIMENT

An extensive set of experiments on varies Chinese NER datasets are conducted to demonstrate the effectiveness of our proposed model. In the following sections, we will list the details of datasets,

Table 1: Data splitting for each dataset

Dataset	Type	Train	Dev	Test
OntoNotes4	Original	15724	4301	4346
	Max length 256	15729	4301	4347
MSRA	Original	46364	-	4365
	Split Dev Data	42000	4364	4365
	Max length 256	42080	4370	4375
Weibo	Original	1350	270	270
	Max length 256	1350	270	270
Resume	Original	3821	463	477
	Max length 256	3821	463	477

parameter settings as well as experimental results and conclude result analysis. Standard precision (P), recall (R) and micro F1-score (F1) are used as evaluation metrics.

#### 4.1 EXPERIMENTAL SETTINGS

##### 4.1.1 DATA DESCRIPTION

The four datasets in our experiments namely OntoNotes 4 Weischedel et al. (2011), MSRA Levow (2006), Weibo Peng & Dredze (2015) He & Sun (2016) and Resume Zhang & Yang (2018) are widely used to evaluate the performances of Chinese NER. The OntoNotes4 dataset carries precise annotation from the news field which contains 4 classes: PER (Person), ORG (Organization), LOC (Location) and GPE (Geo-Political Entity). The training, development and test splits of the OntoNotes4 datasets we adopted in our experiments are the same with Che et al. (2013). MSRA NER dataset of SIGHAN Bakeoff 2006 is another precisely annotated sources from the news field with a larger amount of data. The training set and test set are provided directly with 3 classes: PER, ORG, LOC. To make a development set, several data from the training set are randomly picked up in our experiments. Details for data splitting of MSRA are shown in Table 1. For user generated data, a small dataset extracted from Sina Weibo is considered. The Weibo dataset is annotated with four entity types: PER (Person), ORG (Organization), LOC (Location) and GPE (Geo-Political Entity) and is already divided into training, development and test sets. The resume dataset is released by Zhang & Yang (2018). With the success of the Lattice model, the Resume dataset has also become popular experimental dataset in many papers. It is annotated with eight types of named entities: CONT (Country), EDU (Educational Institution), LOC, PER, ORG, PRO (Profession), RACE (Ethnicity Background) and TITLE (Job Title).

Especially, since our model is conducted on the basis of BERT, the maximum number of input characters is limited. To keep a balance between the computing resources and maintaining the original data, we set the maximum number of inputting characters as 256, where data with exceeded numbers of input characters are cut into shorter sentences. The statistics of the datasets are shown in Table 1, where the original total number of sentences for each data set as well as the total number of sentences after cutting are both given. It should be noticed that in order to obtain a fair verification, the test results of the split sentences are merged together to produce the final result.

##### 4.1.2 IMPLEMENTATION DETAILS

We use the 12-layer pre-trained BERT as the base model to generate the classification embeddings for each entity class and to conduct classification attention. We set the word embedding size of the entity class to 768, the same with the BERT’s hidden size. The classification attention layer is set to 1. The maximum length of the input sentences is set to 256 and the training batches are set to 16. Other shared Hyper-parameters defined in the configuration json file along with the released BERT-Base model remain unchanged. We use BERT-Adam to optimize all the trainable parameters. Learning rate is set to  $5 \times 10^{-5}$  initially and decays during training at a rate of 0.01.

#### 4.1.3 EXPERIMENTAL RESULTS

In this section, we will give the experimental results of our proposed model and previous state-of-the-art methods on Weibo dataset, Chinese Resume dataset, OntoNotes 4 dataset and MSRA dataset, respectively. To exclude the performance improvements from BERT’s own mechanism, we design 2 Baselines based on BERT along with the Classification Attention method, namely BERT + CRF model, BERT with 13 hidden layers (i.e. one additional layer to exclude the improvements of self-attention itself) + CRF model. We treat NER as a sequential labeling problem and adopt BMOES tagging style in this paper since it has been shown that models using BMOES are remarkably better than BIO Yang et al. (2018)

#### 4.1.4 WEIBO DATASET

Table 3 shows the comparisons on Weibo Dataset. We compare our proposed model with the latest models on Weibo dataset. It could be observed that our proposed model achieves state-of-the-art performance. In the first block of Table 3, we report the performances of our baselines. Our baseline BERT-base + CRF achieves an F1-score of 71.05 and BERT with 13 hidden layers + CRF achieves an F1-score of 69.14. In the second block of Table 3, the performances of 3 latest models for Chinese NER are given where Zhang & Yang (2018) and Liu et al. (2019) utilized external resources including predefined lexicon and pre-trained word embeddings. The lattice structure proposed by Zhang & Yang (2018) achieves F1-score of 58.79. By excluding the no shortcut paths between characters, Liu et al. (2019) improves the F1-score to 59.84. Zhu et al. (2019) does not rely on any external resources and achieves the F1-score of 59.31. Compared with the baseline model, Zhang & Yang (2018), Liu et al. (2019) and Zhu et al. (2019) are built on the basis of LSTM of which the feature extraction ability is relatively pooler than Transformer. The performance of our proposed model is presented in the last line in Table 3. Since the F1-scores of the 2 baseline models are around 70, it is assumed that the gap between the training data and the test data is relatively large. So, the entity embeddings obtained from the training data of a certain class may also get relatively large distribution differences with the test class. Hence, the weight for direction removal should be aligned with a larger weight. By doing so, most of the divergences in the distribution of the embeddings in the training set can be excluded, while the common component can be retained. It can be recognized from the result that both the Precision score and Recall score of our model are all higher than the 2 base-line model which demonstrate the effectiveness of our proposed model.

Table 2: Results of Weibo Dataset

Models	P	R	F1
BERT + CRF	70.38	71.74	71.05
13-layers BERT + CRF	70.71	67.63	69.14
Zhang & Yang (2018)	-	-	58.79
Liu et al. (2019)	-	-	59.84
Zhu, Wang and Karlsson (2019)	55.38	62.98	59.31
Our model	<b>71.16</b>	<b>73.91</b>	<b>72.51</b>

Table 3: Results of OntoNotes Dataset

Models	P	R	F1
BERT + CRF	79.08	80.21	79.64
13-layers BERT + CRF	74.06	<b>84.15</b>	78.78
Zhang & Yang (2018)	76.35	71.56	73.88
Liu et al. (2019)	76.09	72.85	74.43
(Zhu, Wang and Karlsson 2019)	75.05	72.29	73.64
Sun et al. (2019a)	-	-	<b>80.21</b>
Zhao et al. (2018)	76.69	71.91	74.22
Our model	<b>79.23</b>	80.92	80.07

#### 4.1.5 ONTONOTES DATASET

Table 4 shows the comparisons on OntoNotes 4 dataset. We compare our proposed model with the latest models on OntoNotes dataset. In the first block of Table 4, we report the performance of the baseline models. Our baseline BERT-base + CRF achieves an F1-score of 79.64 and BERT with 13 hidden layers + CRF achieves an F1-score of 78.78.

In the second block of Table 4, the performances of the 5 latest models are given. Zhang & Yang (2018), Liu et al. (2019), Zhu et al. (2019) and Zhao et al. (2018) are developed based on LSTM whereas Sun et al. (2019a) utilized the character embeddings of BERT. For external resources, Zhang & Yang (2018), Liu et al. (2019) and Zhao et al. (2018) utilized predefined lexicon embeddings and Sun et al. (2019a) utilized pre-trained English NER model. It can be illustrated from the result in

Table 4, Zhu et al. (2019) with no external resources achieves F1-score of 73.64, Zhang & Yang (2018) with lattice structure achieves F1-score of 73.88. Zhao et al. (2018) and Liu et al. (2019) achieve 74.22 and 74.43 respectively. The performance of our proposed model is shown in the last line in Table 4. Since the F1-scores of the 2 baseline models are around 79, the weight for direction removal should also be aligned with a larger weight. It can be recognized from the result that our model got the highest Precision score and competitive F1 score without requiring external resources.

#### 4.1.6 RESUME DATASET

Table 5 shows the comparisons on Resume dataset. We compare our proposed model with the latest models on Resume dataset. In the first block of Table 5, we report the performances of the baseline models. Our baseline BERT-base + CRF achieves an F1-score of 95.76 and BERT with 13 hidden layers + CRF achieves an F1-score of 95.97. In the second block of Table 5, the performances of 5 latest model are given. Liu et al. (2019), Zhu et al. (2019) and Zhang & Yang (2018) are all built on the basis of LSTM. For external resources, Liu et al. (2019) and Zhang & Yang (2018) utilized predefined lexicon and pre-trained word embeddings. It can be told from the result in Table 5, Zhu et al. (2019) with no external resources achieves F1-score of 94.94, Zhang & Yang (2018) with lattice structure achieves F1-score of 95.21. By excluding the no shortcut paths between characters, Liu et al. (2019) improves the F1-score to 95.21. The performance of our proposed model is given in the last line in Table 5. Since the F1-scores of the 2 baseline models are around 95, it is assumed that the gap between the training data and the test data is relatively small. So, the entity embeddings obtained from the training data of a certain class may also get relatively small distribution differences with the test class. Hence, the weight for direction removal should be aligned with a smaller weight. It can be told from the result that our model got both competitive Precision score and Recall score than the Baseline model and forms the highest F1-score finally.

Table 4: Results of Resume Dataset

Models	P	R	F1
BERT + CRF	<b>95.94</b>	95.58	95.76
13-layers BERT + CRF	95.45	<b>96.50</b>	95.97
Liu et al. (2019)	95.27	95.15	95.21
(Zhu, Wang and Karlsson 2019)	95.05	94.82	94.94
Zhang & Yang (2018)	94.81	94.11	94.46
Our model	95.73	96.38	<b>96.06</b>

Table 5: Results of MSRA Dataset

Models	P	R	F1
BERT + CRF	94.98	<b>94.64</b>	94.81
13-layers BERT + CRF	94.91	94.24	94.57
Zhang & Yang (2018)	93.57	92.79	93.18
Liu et al. (2019)	94.58	92.91	93.74
(Zhu, Wang and Karlsson 2019)	93.53	92.42	92.97
Zhao et al. (2018)	93.66	93.05	93.35
Our model	<b>95.20</b>	94.49	<b>94.86</b>

#### 4.1.7 MSRA DATASET

Table 6 shows the comparisons on MSRA dataset. We compare our proposed model with the latest models on MSRA dataset. In the first block of Table 6, we report the performances of the baseline models. Our baseline BERT-base + CRF achieves an F1-score of 94.81 and BERT with 13 hidden layers + CRF achieves an F1-score of 94.57.

In the second block of Table 6, the performances of 4 latest model are given. Zhang & Yang (2018), Liu et al. (2019), Zhu et al. (2019) and Zhao et al. (2018) are all built on the basis of LSTM. For external resources, Zhang & Yang (2018), Liu et al. (2019) and Zhao et al. (2018) utilized predefined lexicon and pre-trained word embeddings. It can be told from the result in Table 6, Zhu et al. (2019) with no external resources achieves F1-score of 92.97, Zhang & Yang (2018) with lattice structure achieves F1-score of 93.18. Zhao et al. (2018) and Liu et al. (2019) achieve 93.35 and 93.74 respectively. The performance of our proposed model is given at the last line in Table 6. Since the F1-scores of the 2 baseline models are around 90, the weight for direction removal should be aligned with a smaller weight. It can be told from the result that our model got competitive Recall score and the highest Recall score and forms the highest F1-score finally.

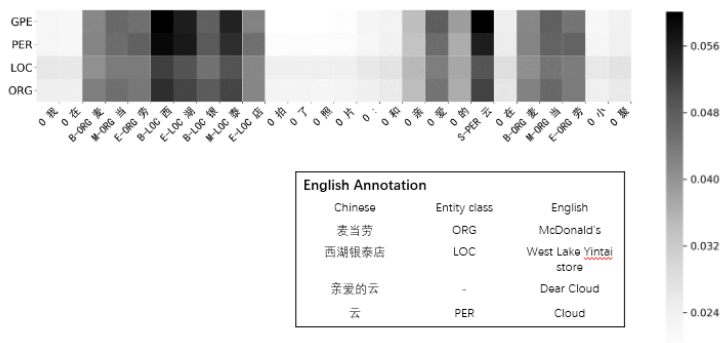


Figure 3: Entity-sentence relation heatmap of the case in Weibo Dataset

#### 4.1.8 CASE STUDY

To explore the process of the Classification Attention, a case study of Weibo Dataset will be shown in this section. We pick a real test sentence from Weibo. The four entity classification embeddings are extracted from the training data which do not contain entities in the test case sentence. The model is trained following the hyperparameters described in the Implementation Details section. The English translation of the input sentence is:

I took a photo at McDonald’s West Lake Yintai store: Gather with Dear Cloud at McDonald’s.

An entity-sentence relation heatmap is shown in Figure 3, where the strength of the relationship is proportional to the depth of the color. The y coordinate shows the categories of different entities and the x coordinate shows the input Chinese character as well as their corresponding labels.

According to the ground truth, most entities are accurately given greater weight after attention though the example sentence contains several irregular symbols. One exception is that only "Cloud" was marked as class "PER" in the ground truth but the whole term "Dear cloud" may also be seen as a whole username that belongs to class "PER".

It should also be noticed that though our method presents indication of all potential entities, the weight of each entity class is not ideally proportional to the ground truth label. West Lake Yintai store is assigned with a greater likelihood in "GPE" than "LOC". At the same time, the organization McDonald’s is assigned with equal weight in "GPE", "PER" and "ORG".

## 4.2 RESULTS ANALYSIS

Our proposed model outperforms previous work on MSRA, Weibo and Chinese Resume dataset and gains competitive results on OntoNotes 4 datasets without using any external resources. The experiments results demonstrate the effectiveness of our proposed model. Compared to Baseline BERT + CRF, the performance improvement after adding Classification Attention mechanism verifies that entity classification embeddings obtained by our model can give extra information to locate entities. Moreover, compared to Baseline BERT with 13 layers + CRF, the better performance of our model excludes the benefits from deeper self-attention layer.

## 5 CONCLUSION

In this paper, we seek to find an entity classification-assisted model for Chinese NER by investigating a mechanism to generate classification embeddings for each entity class using only the character-based BERT pre-trained model and designing a Classification Attention for Chinese NER. It observes a series of weighted representation of the input sentences so that characters similar to a particular entity class could have greater feature weights. In future work, we plan to conduct deeper research on classification embeddings generation and model structure improvements.



## REFERENCES

- Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. 2016.
- Wanxiang Che, Mengqiu Wang, Christopher D Manning, and Ting Liu. Named entity recognition with bilingual constraints. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 52–62, 2013.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- Xiaocheng Feng, Xiachong Feng, Bing Qin, Zhangyin Feng, and Ting Liu. Improving low resource named entity recognition using cross-lingual knowledge transfer. In *IJCAI*, pp. 4071–4077, 2018.
- Hangfeng He and Xu Sun. F-score driven max margin neural network for named entity recognition in chinese social media. *arXiv preprint arXiv:1611.04234*, 2016.
- Masahiro Kaneko and Mamoru Komachi. Multi-head multi-layer attention to deep language representations for grammatical error detection. *arXiv preprint arXiv:1904.07334*, 2019.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv:1901.08746*, 2019.
- Gina-Anne Levow. The third international chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pp. 108–117, 2006.
- Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *arXiv preprint arXiv:1812.09449*, 2018.
- Wei Liu, Tongge Xu, Qinghua Xu, Jiayu Song, and Yueran Zu. An encoding strategy based word-character lstm for chinese ner. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2379–2389, 2019.
- Nanyun Peng and Mark Dredze. Named entity recognition for chinese social media with jointly trained embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 548–554, 2015.
- Nanyun Peng and Mark Dredze. Multi-task domain adaptation for sequence tagging. *arXiv preprint arXiv:1608.02689*, 2016.
- Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.
- Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pp. 1–40. Association for Computational Linguistics, 2012.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. Technical report, Technical report, OpenAI, 2018.
- Marek Rei, Gamal KO Crichton, and Sampo Pyysalo. Attending to characters in neural sequence labeling models. *arXiv preprint arXiv:1611.04361*, 2016.
- Sebastian Ruder<sup>12</sup>, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Sluice networks: Learning what to share between loosely related tasks. *stat*, 1050:23, 2017.

- Erik F Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*, 2003.
- Linghao Sun, Huixiong Yi, and Huanhuan Chen. Back attention knowledge transfer for low-resource named entity recognition. *arXiv preprint arXiv:1906.01183*, 2019a.
- Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*, 2019b.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.
- Chunqi Wang and Bo Xu. Convolutional neural network with word embeddings for chinese word segmentation. *arXiv preprint arXiv:1711.04411*, 2017.
- Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Martha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, Robert Belvin, et al. Ontonotes release 4.0. *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium*, 2011.
- Jie Yang, Shuailong Liang, and Yue Zhang. Design challenges and misconceptions in neural sequence labeling. *arXiv preprint arXiv:1806.04470*, 2018.
- Zhilin Yang, Ruslan Salakhutdinov, and William Cohen. Multi-task cross-lingual sequence tagging from scratch. *arXiv preprint arXiv:1603.06270*, 2016.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.
- Yue Zhang and Jie Yang. Chinese ner using lattice lstm. *arXiv preprint arXiv:1805.02023*, 2018.
- Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*, 2019.
- DongYang Zhao, JiuMing Huang, and Yan Jia. Chinese name entity recognition using highway-lstm-crf. In *Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence*, pp. 64. ACM, 2018.
- Yuying Zhu, Guoxin Wang, and Börje F Karlsson. Can-ner: Convolutional attention network for chinese named entity recognition. *arXiv preprint arXiv:1904.02141*, 2019.