Integrated Char-Word Embedding on Chinese NER using Transformer

Jin ChunGuang, Inwhee Joe*
Hanyang University
guang11644331@gmail.com, *iwjoe@hanyang.ac.kr

트랜스포머를 이용한 중국어 NER 관련 문자와 단어 통합 임배딩

김춘광, 조인휘*
한양대학교 컴퓨터소프트웨어학과
guang11644331@gmail.com, *iwjoe@hanyang.ac.kr

Abstract

Since the words and words in Chinese sentences are continuous and the length of vocabulary is huge, Chinese NER(Named Entity Recognition) always based on character representation. In recent years, many Chinese research has been reconsidered how to integrate the word information into the Chinese NER model. However, the traditional sequence model has complex structure, the slow inference speed, and an additional dictionary information is needed, which is difficult to implement in the industry. The approach in this paper has the state of the art and parallelizable, which is integrated the char-word embeddings, so that the model learns word information. The proposed model is easy to implement, and outperforms traditional model in terms of speed and efficiency, which is improved f1-score on two dataset.

1. Introduction

The NER(Named Entity Recognition) task is one of the downstream tasks in the field of NLP, and it is also an essential work before many projects, such as knowledge graph, question and answer. There are two main problems with NER in Chinese. First, word segmentation. In Chinese sentences, words are continuous, unlike English words, which are easily separated by spaces. Although the current word segmentation technology is very excellent, there are a large number of special words in special local names in specified fields medicine or finance, it is still unable to achieve perfect word segmentation. Second, the Chinese vocabulary is too large, resulting in a large vocabulary length, which affects the speed and effect of the model. Therefore, the input on Chinese NER commonly using char-based embeddings.

Recently, The Lattice-LSTM[1] brought Chinese NER to a new level. It designed a box structure to insert vocabulary information between the word information, and the word information was extracted from the dictionary they built and modified the structure of LSTM integrates the word information statically into the model, which significantly improves the performance of the model. But LSTM itself is difficult to parallelize, the model training and inference speed is very slow, the structure is modified to incorporate new information, resulting in the model structure being too complex, the efficiency is greatly reduced, and the information of the newly added word is only the current word the last word of can be learned, the previous words cannot be learned, and the directionality is also problematic, even if it does improve the score of the mode.

Since the emergence of transformer[2], the pre-training model almost dominates all field in NLP, which is an improvement of the traditional seq2seq[3] structure. in this paper we using BERT[4] model which improved by the encoder part of the Transformer. It is composed of a multi-head self-attention layer and a fully connection layer. It is bidirectional and does not suffer from the lack of long-terms dependence in RNN[5]. As an encoder model, its output is a dynamic word embedding that learns contextual information well. At the same time, self-attention perfectly supports parallelization. Compared with the LSTM model, both the effect and the speed of inference are greatly improved. Use its input to access the specified model for custom downstream tasks.

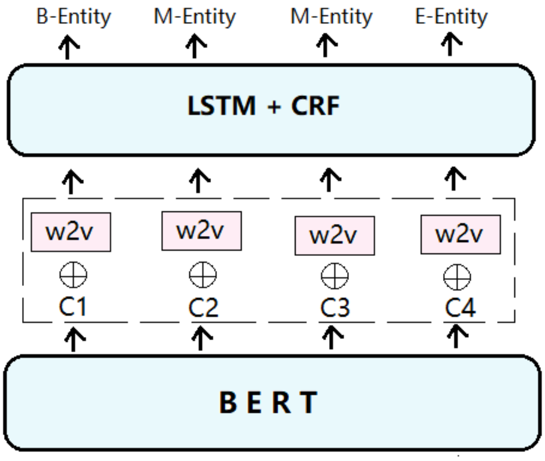

Figure 1. Model Structure

The approach in this paper is using char indices input into the BERT model (Figure 1), the output char embeddings learns the context information, and then integrate it with the pre-trained word vector in advance. Now, each character of the entity is integrated the vocabulary information and obtained a new char-word embeddings. And then input it into the standard LSTM+CRF[6] layer for final classification. The output of the BERT, which supports parallelization and learns the context information, also add the word vector incorporating new information, improved the performance of Chinese NER more efficiently.

2. Methodology

2.1 Dimensional Reduction

First, use the input based on word embedding like other Chinese NER tasks. After passing the self-attention and fully connection layer of BERT, the output is a 768-dimensional char embeddings(Figure 1). In order to better integrate the vocabulary information, the output char embeddings must be reduced in dimension. in this paper we using a training approach to reduce the char embeddings, that is, a hidden unit is used as a 300-dimensional of fully connected layer, the activation function used ReLU[7] to prevent the vanishing gradient. Then we get a 300-dimensional char embeddings (Formula 1), that c represents the char vector, v(i), v(j) represent the entity part, and e represents the output sentence embedding encoded by BERT.

$$e = \{c_1, c_2, \dots v_i, v_j, \dots c_n\} , \quad (0 < i < j < n)$$

(1)

2.2 Concatenate

The word vector used in this paper is Baidu Encyclopedia 300d[8], which is a word vector trained through the Chinese analogy inference task. The training corpus is Size=4.1G, tokens=745M, |V|=5422k. The next step is to integrate the word information. There are many options, such as max pooling, dot product, or training a fully connected layer like the attention in seq2seq. But we finally chose to directly added the word vector to the character embeddings. The difference with Lattice-LSTM is that we added the word of the entity to each character that composes the entity, so that each word incorporates the entity's information (Formula 2) instead of just the last character. w represents the word vector, v is the entity's character vector and word vector added together. In this way, we get a new embedding matrix E which is s char-word embeddings(Formula 3). Just enter it into the standard NER model.

$$\boldsymbol{V_i} = \{\boldsymbol{v_i} \oplus \boldsymbol{w_i}\} \qquad (2)$$

$$E = \{c_1, c_2, \dots V_i, V_j, \dots c_n\} \qquad (3)$$

3. Experiment

3.1 Dataset and Metrics

The dataset is using Weibo[9] and R esume[1], and the information is shown in the Table 1. The evaluation standard using the macro average F1-Score.

Dataset	Type	Train	Dev	Test
Weibo	Sentence Char	1.4k 73.8k	0.27k 14.5k	0.27k 14.8k
Resume	Sentence Char	3.8k 124.1k	0.46k 13.9k	0.48k 15.1k

Table 1. Dataset

3.2 Pre-trained Model

We using the BERT model in this paper is the Chinese whole word mask(WWM) BERT[10], which is an upgraded version of BERT released by Google on May 31, 2019. It mainly changes the training samples of the original pre-training stage. Generate strategy. Simply, the original word segmentation approach based on word-piece will divide a very long word into root and affix. When generating training samples, these separated subwords will be randomly masked. But in WWM BERT, if part of the word-piece sub-word of a complete word is masked, the other parts of the same word will also be masked, that is, the whole word mask means.

3.3 Result

The experimental result(Table 2) shows that the performance of the model is improved after the entity is integrated with the char-word information, mainly on the Weibo dataset, but the improvement on the resume dataset is not particularly obvious, because there are many netizens on Weibo platform. The terminology is more folk and unofficial, and has many special words. Therefore, we speculate that the approach in this paper is more suitable for datasets with more rare words.

Model / Data	Weibo	Resume
Lattice-LSTM	58.79	94.46
BERT	66.73	95.30
BERT Char-Word Embed	**68.84**	**96.47**

Table 2. The F1-Score of Result

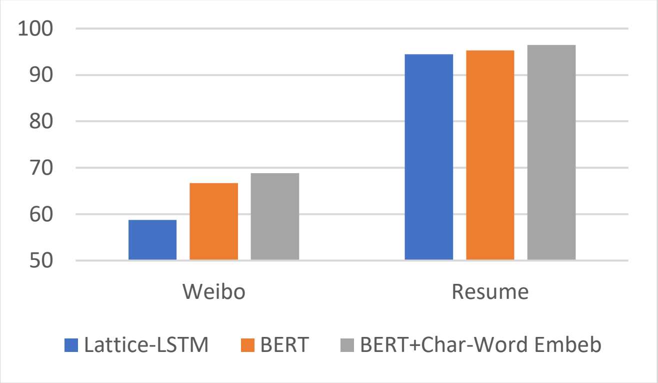

Figure 2. The F1-Score of Result

4. Relative Work

4.1Lattice-LSTM

There are two main advantages to Lattice-LSTM[1]. First, it preserves all the possible lexicon matching results that are related to a character, which helps avoid the error propagation problem introduced by heuristically choosing a single matching result for each character. Second, it introduces pre-trained word embeddings to the system, which greatly enhances its performance.

5. Conclusion and Future Work

In this paper, we introduced a approach of integrated the char-word information, and effectively improved the lack of vocabulary information in Chinese NER. And also using the new architecture of Transformer, which greatly improves the training speed compared to the traditional sequence model. Since the approach in this paper is easy to implement, we will also use it on other large-scale pre-training models and compare the differences between with them.

Reference

[1] Zhang Y , Yang J . Chinese NER Using Lattice LSTM[J]. 2018.

[2] Vaswani A , Shazeer N , Parmar N , et al. Attention Is All You Need[J]. arXiv, 2017.

[3] Bahdanau D , Cho K , Bengio Y . Neural Machine Translation by Jointly Learning to Align and Translate[J]. Computer Science, 2014.

[4] Devlin J , Chang M W , Lee K , et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. 2018.

[5] Hochreiter S . The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions[J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 1998, 06(2):-.

[6] Huang Z , Wei X , Kai Y . Bidirectional LSTM-CRF Models for Sequence Tagging[J]. Computer Science, 2015.

[7] Agarap A . Deep Learning using Rectified Linear Units (ReLU). 2018.

[8] Li S , Zhao Z , Hu R , et al. Analogical Reasoning on Chinese Morphological and Semantic Relations. 2018.

[9] He H , Xu S . F-Score Driven Max Margin Neural Network for Named Entity Recognition in Chinese Social Media[J]. 2016.

[10] Cui Y , Che W , Liu T , et al. Pre-Training with Whole Word Masking for Chinese BERT. 2019.