Named Entity Recognition Model

Shuquan Zhao

March 5, 2021

1 Data Preprocessing

The data preprocessing technique used here is to convert sentence strings and NER strings into tokens. According to the characteristics of the datasets, each word is separated by a blank character, so the blank character can be used as a delimiter. The sentences in the datasets are converted into token arrays. At the same time, the NER string uses the same operation to convert the NER string into a NER tag array. The conversion of sentence strings into token arrays is because this model encodes sentences at the word level, so it needs to read each word in the sentence and then get each word embedding vector. The reason why NER strings are converted into tag arrays is because NER tags need to be encoded separately for model training, prediction and evaluation.

2 Input Embedding

The Input Features tried in this model are ”word”, ”word.lower()”, ”word[-3:]”, ”word[-2:]”, ”word.isupper()”, ”word .istitle()”, ”BOS”, ”EOS”. The ”word” feature is the word2vec embedding of the word. ”Word.lower()” is the word2vec embedding that converts the word to lower case before getting the word. ”Word[-3:]” is to get the last three characters of word, and then encode these three characters as a token. ”Word[-2:]” is to get the last two characters of word, and then encode these two characters as a token. The word embedding of this model uses a pre-trained model, which can save model training time.

3 NER Model

This model contains two main layers LSTM layer, Attention layer and CRF layer.

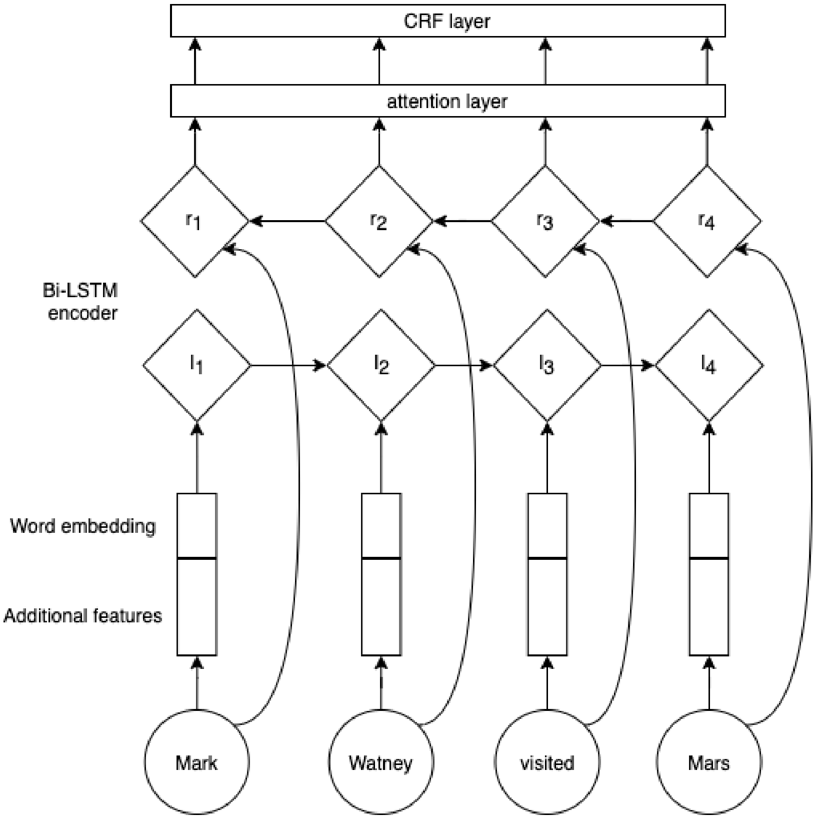

Figure 1: NER Model

The input of BiLSTM-CRF is a word embedding vector, and the output is the predicted label corresponding to each word [1]. Attention mechanism is a technology that allows the model to focus on important information and fully learn and absorb it. It is not a complete model. It should be a technology that can be used in any sequence model. There are types of attention score calculation:

1. Dot Product

   $$a(q, k) = q^T k$$

   This method is more direct and saves even the weight matrix. Establishing the relationship map of q and k directly has the advantage of faster calculation speed and no parameters, which reduces the complexity of the model. But the dimensions of q and k need to be the same.

2. Scale Dot Product

   One problem with the dot product method above is that as the vector dimension increases, the final weight will also increase. In order to improve the calculation efficiency and prevent data overflow, it is scaled [2].

$$a(q, k) = \frac{q^T k}{\sqrt{|k|}}$$

The CRF layer can add some constraints to ensure that the final prediction result is valid. These constraints can be automatically learned by the CRF layer during training data. The loss function in the CRF layer includes two types of scores: emission score and transition score. Emission score comes from the output of the middle BiLSTM layer. Transition score is a parameter of the BiLSTM-CRF model. It is initialized randomly before training the model. These scores will be updated as the training iterates, and the CRF layer can learn these constraints by itself.

4 Evaluation

4.1 Evaluation setup

4.1.1 Learning Rate

The initial value of learning rate is 0.01, and then the learning rate scheduler is used to dynamically adjust the learning rate. The configuration of Learning rate schuduler is:

mode	max
patience	3
verbose	True
threshold	0.001

4.1.2 Optimizer

The optimizer used in this model is Stochastic Gradient Descent (SGD). SGD is a basic optimization method. It needs to repeatedly put the entire set of data into the neural network for training. This consumes a lot of computing resources, but SGD accelerates the training process of the neural network. And it will not lose too much accuracy.

- train_data: Train sentence tokenize dataset. [[word1, word2, word3, ...], [], ...]
- val_data: Validation sentence tokenize dataset.
- test_data: Test tokenize dataset.
- features: Input features list. value: ['word', 'BOS', 'EOS'].
- word_emb_model: Word embedding model name, like "word2vec-google-news-300".

- word_emb_size: Word embedding model dim.
- epochs: Control the number of model trainings. Value: 20.

4.2 Evaluation result

4.2.1 Performance Comparison

This part builds different CRF models and then compares the performance of these models. Here I tried to build BiLSTM-CRF, Traditional CRF and spaCy CRF. The validation F1 score of the best model is 0.97814.

Models	Validation F_1
BiLSTM CRF (Baseline Model)	0.94124
best model	0.97814
CRF	0.91214
spaCy CRF	0.92225

4.2.2 Ablation Study - different embedding model

In this part, we construct different input embedding and then compare the performance of the model. The combinations tried here are "only word2vec", "word2vec + BOS + EOS", "word2vec + BOS + EOS + word[-3:] + word[-2:]", "Word2vec + BOS + EOS + word.isupper() + word.istitle()".

Embedding Models	Validation F_1
Only word2vec	0.9243
Word2vec + BOS + EOS	0.9075
Word2vec + BOS + EOS + word[-3:] + word[-2:]	0.7609
Word2vec + BOS + EOS + word.isupper() + word.istitle()	0.9046

4.2.3 Ablation Study - different layer strategy

In this part, try to use different layers to find the optimal number of model layers. Through experiments, it is found that different model layers will affect the performance of the model.

LSTM Layers	Validation F1
1	0.9412
2	0.8661

References

[1] Huang, Z., Xu, W., Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991.

[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).