Vanilla BiLSTM-FNN Model for Named Entity Recognition

Zeqiang Lai
Department of Computer Science
Beijing Institute of Technology
1120161865@bit.edu.cn

Jinxuan Jin
Department of Computer Science
Beijing Institute of Technology
1120161864@bit.edu.cn

Wenzhuo Liu
Department of Computer Science
Beijing Institute of Technology
1120161868@bit.edu.cn

Tian Huan
Department of Computer Science
Beijing Institute of Technology
1120161861@bit.edu.cn

Anteng Li
Department of Computer Science
Beijing Institute of Technology
1120161866@bit.edu.cn

Xueyan Guo
Department of Computer Science
Beijing Institute of Technology
1120162336@bit.edu.cn

Abstract

We propose a pure neural network architecture for text-tagging based bidirectional LSTM(long short term memory) and FNN(feedforward neural network). Our method is an end-to-end approach, which does not involve any language dependent information, such as part of speech tags, chunking tags. We only use word embeddings to encode the input words. We test our model on an english named entity recognition task. We show that our vanilla network is able to achieve a fairly well performance.

1 Introduction

Named entity recognition is a challenging learning problem in the field of natural language process. The main task of it is to determine whether a phrase in the sentences is a named entity which contains the names of persons, organizations and locations. Besides, the model should be able to indicate which type of named entity it is.

For this task, there are numbers of approaches have been developed. Conditional Random Field(CRF;Lafferty et al, 2001)[6] is widely used and there are many variants of it, such as Softmax-margin CRFs[3]. Neural approaches using LSTM(Hammerton, James) [4] or CNN are also popular. Moreover, some models combine these together. Chiu .et.al. [1] 's model is a integration of LSTM and CNN. There are also works on LSTM and CRF[7][5]. And Ma, Xuezhe .et.al. go a step further joining LSTM,CNN and CRF all together.

Our method is inspired by Lample et.al.'s work [7]. For simplicity, we skip the CRF layer and character level embeddings. We only use neural networks, LSTM and FNN to build our model.

The rest of the paper is organized as follow: Section2 illustrate the architecture of our model in detail. In Section3, experiments are carried out and the computational results are described. Finally, we briefly conclude and discuss some directions for future work in Section 4.

2 Architecture

2.1 Embedding layer

We used pretrained word embedding to encode the input words. Embedding are pretrained using GloVe[8], an unsupervised learning algorithm for obtaining vector representations for words. Word embeddings with different dimensions are tested, respectively. And we observe a significant improvement as the dimension increases, see table 1.

2.2 BiLSTM layer

A bidirectional LSTM layer is adopted to capture the information of front words and back words. The output of BiLSTM is the concatenation of the outputs of forward LSTM and backward LSTM. See equations.

$$\overrightarrow{h}_i = LSTM(\overrightarrow{h}_{i-1}, E_i)$$
$$\overleftarrow{h}_i = LSTM(\overleftarrow{h}_{i+1}, E_i)$$
$$C_i = [\overrightarrow{h}_i, \overleftarrow{h}_i]$$

2.3 FNN layer

Feedforward neural network(FNN) layer is a fully connected layer that process the information of words and their context. In our model, we stack two of it with 100 neurons to produce the final features.

2.4 Output layer

The final features outputted by FNN layer are sent to a Softmax classifier to generate predictions.

$$O_i = Softmax(Z_i^{(2)})$$

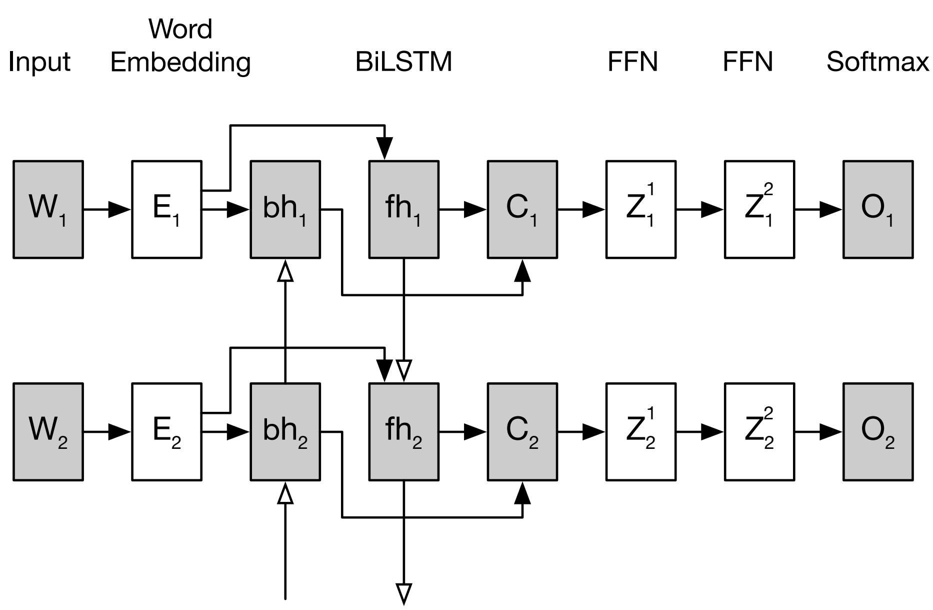

Figure 1: BiLSTM-FNN model for sequence labeling.

3 Experiment

3.1 Dataset

We used CoNLL-2003 datasets[9] that contain independent English entity labels for English. The dataset contain four different types of named entities: locations, persons, organizations, and miscellaneous entities that do not belong in any of the three previous categories. In order to build an end-to-end model, we only used the words and their named entity tags.

3.2 Training

We implemented our model using Keras [2] and run the experiments on a MacBook Pro with one i7 Core CPU. Network parameters are learned using stochastic gradient descent(SGD), augmented with the Adam optimizer.

The dimensions of forward and backward LSTM are set to 100. And word embeddings with different dimensions(50,100) were adopted to see the impact of embeddings on the performance. We also test dropout during training.

3.3 Results

After training on training set for 15 epochs. We get the results using different configurations. See Table 1.

The origin model uses fixed word embedding and doesn't perform dropout. Comparing with this model, the one that use trainable embedding has a significant improvement on performance. This can be interpreted as a lack of part of speech features in general word embeddings, which might be useful to distinguish entities(for example, a entity is usually a noun). And if we increase the dimension of word embedding, we get a even better performance due to the similar reason.

Dropout is useful for LSTM layer but not for embedding layer. We assume that dropout of embedding layer causes too much loss of words information.

Table 1: English NER results(CoNLL-2003 test set) with our model using different configurations.

Experiments	F1 Score
origin	54.10
trainable embedding(50d)	66.47
dropout lstm	70.29
dropout embedding	67.59
trainable embedding(100d)	74.57
text8 embedding(128d)	73.45

4 Conclusion

We have proposed a simple neural based model for named entity recognition task. Our experiments show that our model is able to achieve a fairly well performance evaluated by F1 metric.

Several directions remain open for future research. First, as shown by Guillaume Lample et al.[7] we can combine CRF and character embedding with our model to get improvement. Further, can our model work well on other datasets or in real-world situations. Finally, Is there a possibility to even use attention mechanism to get better result?

References

[1] Jason PC Chiu and Eric Nichols. "Named entity recognition with bidirectional LSTM-CNNs". In: *arXiv preprint arXiv:1511.08308* (2015).

[2] François Chollet et al. *Keras*. `https://keras.io`. 2015.

[3] Kevin Gimpel and Noah A Smith. "Softmax-margin CRFs: Training log-linear models with cost functions". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics. 2010, pp. 733–736.

[4] James Hammerton. "Named entity recognition with long short-term memory". In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics. 2003, pp. 172–175.

[5] Zhiheng Huang, Wei Xu, and Kai Yu. "Bidirectional LSTM-CRF models for sequence tagging". In: *arXiv preprint arXiv:1508.01991* (2015).

[6] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289. ISBN: 1-55860-778-1. URL: `http://dl.acm.org/citation.cfm?id=645530.655813`.

[7] Guillaume Lample et al. "Neural architectures for named entity recognition". In: *arXiv preprint arXiv:1603.01360* (2016).

[8] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "GloVe: Global Vectors for Word Representation". In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1532–1543. URL: `http://www.aclweb.org/anthology/D14-1162`.

[9] Erik F Tjong Kim Sang and Fien De Meulder. "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition". In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics. 2003, pp. 142–147.