Improving a Neural Named Entity Model with Linguistic Factors

Hai Nguyen[1], Long Nguyen[2], Tan Le[3], and Phuoc Tran[4]

[1]Faculty of Physics, University of Pedagogy, Ho Chi Minh City, Vietnam
Email: hainm@hcmup.edu.vn
[2]CLC Lab, Faculty of Information technology, University of Science, Ho Chi Minh City, Vietnam Email: nhblong@gmail.com
[3]Faculty of Computer Science, Université du Québec à Montréal, Québec, Canada
Email: le.ngoc_tan@courrier.uqam.ca
[4]NLP-KD Lab, Faculty of Information technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam Email: tranthanhphuoc@tdt.edu.vn

Abstract. This paper reports the use of word-level linguistic factors to a neural model for named entity recognition tasks. Several word-level linguistic factors, such as lemmas, word clusters, part-of-speech tags, and syntactic chunk tags, were combined in embedding layers. These factors were incorporated into a bidirectional recurrent neural model. The experiments showed that our proposed method obtained a better performance with a significant gain compared to a baseline model. Adding these linguistic factors allowed the neural model work better in case of data sparseness or language ambiguity problems.

Keywords: Named Entity Recognition, Neural Networks, Linguistic Factors

1 Introduction

Named Entities (NEs), especially person names (PER), location names (LOC), and organization names (ORG), have a very important role in many Natural Language Processing (NLP) applications (e.g. information extraction, information retrieval, machine translation). NE recognition (NER) is not only a fundamental task but also one of the most challenging tasks in NLP research area. Since 1995, there have been many NER systems developed for many languages [1] with the domination of supervised learning methods such as Support Vector Machine [2] and Conditional Random Field (CRF) [3].

NER by using neural network models has received much attention in recent years because these models have shown remarkable results and improvements over conventional NER models [4]. These models usually take advantages of using Long Short Term Memory (LSTM) [5] models which improve Recurrent Neural Network (RNN) [6] models by their ability to capture long sequence context of

Phuoc Tran: Corresponding author

sentences. Recent neural NER methods [4], [7], [8] captured the context of a sentence in both directions of the sentence using forward and backward LSTM models. Their methods are called bidirectional LSTM (B-LSTM) and have shown significant improvements in NER tasks. Even there were methods which tried to adapt Convolutional Neural Network models [9] or CRF algorithms [4] to B-LSTM models, they still kept B-LSTM as the most important part. Even these B-LSTM models have shown promising results, they are lack of ability of using linguistic information for surface words. The additional linguistic information can help to resolve language ambiguity or data sparseness problems which usually occur when training data are not large enough.

This paper presents a novel method to incorporate additional linguistic factors at word level including lemmas, word clusters, part-of-speech (POS) tags, and syntactic chunk tags to a B-LSTM model for English NER tasks. The combination of these factors significantly improves the B-LSTM model performance with impressive results on an English data set.

The remainder of this paper is organized as follows. Some related works are introduced in Section 2. The proposed method is presented in Section 3. Then the experiments and the evaluation on the data are reported in Section 4. Finally, Section 5 presents some conclusions and future work.

2 Related Work

Several neural network architectures have been proposed to tackle NER tasks. [10] proposed a CNN model which can learn proper characteristics of words to automatically extract features with a CRF layer on the top. [11] introduced a similar architecture using spelling features. It was later improved with character-level embeddings [4]. [4] also presented an architecture that using a stack of LSTM layers. The above methods used B-LSTM layers as a key part in their architecture augmented with some features such as pre-trained word embedding layers and dropout layers.

3 Proposed Method

In this study, we investigate the possibility to integrate linguistic factors to a neural network model for NER tasks. As an initial work, our purpose is to find how additional linguistic factors can resolve data sparseness and language ambiguity problems in an specific B-LSTM model. Assume that we have a sentence containing N words; each word is augmented with L linguistic factors. Then, we can express the sentence as $\{(\{x^{(n,l)}\}_{l=0}^{L}, y^{(n)})\}_{n=1}^{N}$.

LSTMs take an input of a sequence of vectors $(x_1, x_2, ..., x_n)$ and produce an output of a sequence of vectors $(h_1, h_2, ..., h_n)$ to represent the information at each input step. LSTMs have been designed to incorporate a memory cell which can protect and control the cell state. They use several gates to control the amount of information from the previous states which should be forgotten and the information from the inputs which should be updated to the memory

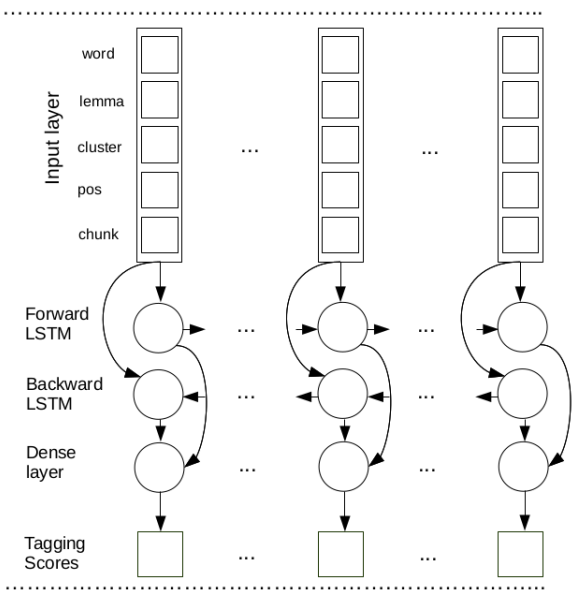

Fig. 1. Our neural network based named entity architecture.

cell [12]. There are many variants of LSTM implementations. In this paper, we reuse the implementation of Lample et al. (2016) [4]:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{1}$$

$$c_t = (1 - i_t) \odot c_{t-1} + i_t \odot tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{2}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{3}$$

$$h_t = o_t \odot tanh(c_t) \tag{4}$$

where σ is the element-wise sigmoid function, and $\odot$ is the element-wise product. c_t and o_t are the cell state and the output at the step t, respectively.

A B-LSTM which contains a forward LSTM and a backward LSTM operate on a sequence in forward and backward directions, respectively. If $\overrightarrow{h_j}$ is the summary representation of a word at position *j-th* from the left of a sequence, $\overleftarrow{h_j}$ will be the summary representation of that word starting from the right of the sequence. It means the combination of $h_t = [\overrightarrow{h_j}, \overleftarrow{h_j}]$ summarizes the representation of the whole sequence.

Figure 1 shows the architecture of our proposed model with integrating linguistic factors to the B-LSTM model. At the beginning, the input layer is a concatenation of several embedding layers (i.e. L layers) encoding the linguistic

factors to word vectors. The B-LSTM layer later represents them as context dependent word vectors $h_t = [\overrightarrow{h_j}, \overleftarrow{h_j}]$. Eventually, these context vectors are inputed to a dense layer which is a regular densely-connected neural network layer with a softmax activation function to produce tagging scores.

4 Experiments

4.1 Training Configuration

We used Keras[1] library to implement all our experiments. The baseline was a simple B-LSTM model with an embedding layer. All neural models were configured with 128 embedding layer dimensions and 100 hidden layer dimensions. For the linguistic factors, we used 64 embedding layer dimensions to encode each linguistic factor including word lemmas, word clusters, POS tags, and syntactic chunk tags. The word lemmas and word clusters were collected by NTLK toolkit[2] and Brown Cluster[3], respectively.

To train our neural networks, we used stochastic gradient descent (SGD) with a fixed learning rate of 0.01. The batch size was 32 and the number of epochs was 200. The brown cluster size was 100.

For evaluation of our neural networks, we used the CoNLL script[4]. To test the statistical significance, we applied bootstrapping resampling koehn:2004:EMNLP to measure the significant level ($p < 0.01$) of F-score differences between the neural models.

4.2 Data sets

We conducted our experiments on CoNLL-2003 English dataset[5]. Because the dataset already contains POS tags and syntactic chunk tags, we only need to add word lemmas and word clusters factors. For the training step, we used the training set containing 14,985 sentences. We evaluated on the *testa* and *testb* testing set. Table 1 provides statistics on the dataset.

Dataset	**#tokens**	**#types**	**#sentences**
train	204,562	23,624	14,985
testa	51,573	9,966	3,464
testb	46,624	9,485	3,682

Table 1. Statistics of the English data set.

[1] https://keras.io/
[2] http://www.nltk.org/
[3] https://github.com/percyliang/brown-cluster
[4] http://www.cnts.ua.ac.be/conll2002/ner/bin/conlleval.txt
[5] http://www.cnts.ua.ac.be/conll2003/ner/

The dataset contains columns separated by a single space. Table 2 illustrates linguistic factors in an English sentence. The first item on each line is a word, the second a lemma, the third a word cluster, the fourth a POS tag, the fifth a syntactic chunk tag and the sixth the NE tag. The NE tag is one of classes including PER, LOC, ORG, or MISC (Miscellaneous).

Word	lemma	word cluster	pos	chunk	NE
Only	only	011000111	RB	I-NP	O
France	france	111001	NNP	I-NP	I-LOC
and	and	0011110	CC	I-NP	O
Britain	britain	111001	NNP	I-NP	I-LOC
backed	back	0110011011	VBD	I-VP	O
Fischler	fischler	0110100	NNP	I-NP	I-PER
's	's	001110	POS	B-NP	O
proposal	proposal	01010111	NN	I-NP	O
.	.	0010	.	O	O

Table 2. Example of linguistic factors.

4.3 Results

As mentioned previously, the aim of the tests was to evaluate the effectiveness of integrating linguistic factors to a B-LSTM model. For that reason, we used a pure B-LSTM model as a baseline. This section presents our comparisons for different experiments on English NER.

Table 3 shows the scores reported in F_1 score. It is clearly to realize that either the individual factor or the combination of factors will improve the performance of the B-LSTM model. The word lemma factor reduced the data sparseness problem in an inflectional language such as English. Other remaining factors resolved the language ambiguity problems. We observed that the combination of all the factors gave us a biggest improvement in overall performance of $(+7.49, +10.74)$. For each factor, the POS factor gave us an particularly increase of $(+4.46, +8.05)$; the word cluster factor marked an increase of $(+2.80, +5.60)$, while the chunk factor resulted in a difference of $(+1.72, +1.36)$ and finally the lemma factor resulted in a increase of $(+1.13, +1.38)$.

Models	testa	testb
B-LSTM (baseline)	74.41	64.92
B-LSTM + lemma	75.54	66.30
B-LSTM + word cluster	77.21	70.52
B-LSTM + pos	78.87	72.97
B-LSTM + chunk	76.13	66.28
B-LSTM + all factors	81.90	75.66

Table 3. NER results on the English dataset.

Even these factors helped to overcome the data sparseness and language ambiguity problems, they could not resolve the difficulty in classifying NEs to their right classes. For example, sometimes, a person name can be used to indicate a location or an organization. In the test set, *Charles de Gaulle* was manually tagged as a LOC name but the model predicted *Charles* in that location phrase as a PER name.

5 Conclusion and Future Work

Prior work has indicated the effectiveness of using B-LSTM models in NER tasks. However, these studies have not focused on the advantages of linguistic factors which can help to solve the language ambiguity and data sparseness problems. In this paper, we studied the importance of additional linguistic factors in a B-LSTM model. We found that in all cases, adding either a linguistic factor or a combination of linguistic factors substantially increases the performance of a B-LSTM model. This study, therefore, proves the benefits of adding linguistic information to a neural network model in NLP research area. In addition, these factors used in our study are easy to establish via existing toolkits. Most notably, this is the first study, according to our knowledge, to investigate the effectiveness of incorporating linguistic factors to a neural network for the English NER tasks.

However, in this paper, we only integrated the linguistic factors to embedding layers by concatenating linguistic vectors. In the future work, we will investigate other possible methods to integrate more effectively these factors . Besides that, we will also examine the possibility of applying additional layers such as CNN layer, CRF layer, or dropout layer into our proposed method.

References

1. David D. Palmer and David S. Day. 1997. A statistical profile of the Named Entity task. In Proceedings of the fifth conference on Applied natural language processing (ANLC '97). Association for Computational Linguistics, Stroudsburg, PA, USA, 190-193. DOI: https://doi.org/10.3115/974557.974585
2. Cortes, C. Vapnik, V. Mach Learn (1995) 20: 273. https://doi.org/10.1007/BF00994018
3. John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proceedings of the Eighteenth International Conference on Machine Learning (ICML '01), Carla E. Brodley and Andrea Pohoreckyj Danyluk (Eds.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 282-289.
4. Lample, Guillaume, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami and Chris Dyer. Neural Architectures for Named Entity Recognition. HLT-NAACL (2016).
5. Felix A. Gers, Jürgen A. Schmidhuber, and Fred A. Cummins. 2000. Learning to forget: Continual prediction with lstm. Neural Comput. 12(10):24512471. https://doi.org/10.1162/089976600300015015.

6. Christoph Goller and Andreas Kchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. In In Proc. of the ICNN-96. IEEE, pages 347352.
7. Jason P. C. Chiu and Eric Nichols. 2015. Named entity recognition with bidirectional lstm-cnns. CoRR abs/1511.08308. http://arxiv.org/abs/1511.08308.
8. Yushi Yao and Zheng Huang. 2016. Bi-directional LSTM recurrent neural network for chinese word segmentation. CoRR abs/1602.04874. http://arxiv.org/abs/1602.04874.
9. Onur Kuru, Ozan Arkan Can, and Deniz Yuret. 2016. Charner: Character-level named entity recognition. In Nicoletta Calzolari, Yuji Matsumoto, and Rashmi Prasad, editors, COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan. ACL, pages 911921. http://aclweb.org/anthology/C/C16/C16-1087.pdf.
10. Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. J. Mach. Learn. Res. 12:24932537. http://dl.acm.org/citation.cfm?id=1953048.2078186.
11. Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. CoRR abs/1508.01991. http://arxiv.org/abs/1508.01991.
12. Sepp Hochreiter and Jrgen Schmidhuber. 1997. Long short-term memory. Neural Comput. 9(8):17351780. https://doi.org/10.1162/neco.1997.9.8.1735.