Enhancing LSTM-based Word Segmentation Using Unlabeled Data

Bo Zheng, Wanxiang Che*, Jiang Guo, Ting Liu
†Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology, China
{bzheng, car, jguo, tliu}@ir.hit.edu.cn

Abstract. Word segmentation problem is widely solved as the sequence labeling problem. The traditional way to this kind of problem is machine learning method like conditional random field with hand-crafted features. Recently, deep learning approaches have achieved state-of-the-art performance on word segmentation task and a popular method of them is LSTM networks. This paper gives a method to introduce numerical statistics-based features counted on unlabeled data into LSTM networks and analyzes how it enhances the performance of word segmentation model. We add pre-trained character-bigram embedding, pointwise mutual information, accessor variety and punctuation variety into our model and compare their performances on different datasets including three datasets from CoNLL-2017 shared task and three datasets of simplified Chinese. We achieve the state-of-the-art performance on two of them and get comparable results on the rest.

Keywords: word segmentation, statistics-based features, neural network, unlabeled data

1 Introduction

Most of the natural language processing tasks are processed in the units of words. In order to do downstream tasks, word segmentation is basic and important in those languages like Chinese and Japanese which are written in continuous sequences of characters, without delimiters between words. For Vietnamese, there are two kinds of white spaces between characters, one is inside words, the other one is between words. The goal of word segmentation for Vietnamese is to recognize these two kinds of white spaces. Word segmentation problem is widely solved as the sequence labeling problem. The traditional way to this kind of problem is machine learning method like conditional random field [5] with hand-crafted features. Neural network-based models have been extensively used in natural language processing during recent years, due to their strong capability of automatical feature learning. LSTM networks is a popular method on word segmentation task. A meaningful way to improve the performance of existing

* Email correspondence.

approaches is introducing more helpful features into the basic model or finding new ways to introduce these existing features.

Previous researchers have tried to use auto-segmented result of large scale unlabeled data [13] or statistical magnitudes like mutual information [6], accessory variety [2] to help the supervised learning system. Performance improvement is achieved in their works. However, those previous works use the statistical results as discrete features while it may cause the problem of missing information. For example, [10] only uses the integer part of the mutual information, which will lose the information of the fractional part. To the best of our knowledge, there hasn't been any work that feeds numerical statistics-based features into LSTM networks. We think it is worth to study since numerical features seem more suitable than discrete features in LSTM networks. In this work, we present an approach that directly uses numerical statistics-based features [10] counted on unlabeled data to enhance word segmentation based on LSTM networks. The statistics-based features we utilize include pointwise mutual information, accessor variety and punctuation variety [10]. We also use pre-trained character-bigram embeddings to replace randomly initialized ones. We conduct our experiments on six datasets including three datasets from CoNLL-2017 shared task and three datasets of simplified Chinese. We achieve the state-of-the-art performance on two of them and get comparable results on the rest of them.

2 Related Work

Neural network approaches are popular in word segmentation task, [9] used a tensor neural network to achieve extensive feature combinations, capturing the interaction between characters and tags. [7] combined semi-CRF with neural network to solve NLP segmentation tasks, their experiments show that their neural semi-CRF model benefits from representing the entire segment. [14] proposed a transition-based neural word segmentation model, they replaced the manually-designed discrete features the neural features in a word-based segmentation framework. Both [14] and [7] used word-level information. [1] proposed a novel neural network framework which thoroughly eliminates context windows and can utilize complete segmentation history.

Using unlabeled data to enhance Chinese word segmentation has also been widely applied. [10] proposed a unified solution to include features derived from unlabeled data to a discriminative learning model based on conditional random field. The feature set includes mutual information, accessor variety, punctuation variety and other statistics-based features. Their experiments are based on conditional random field. Our model uses several features from this paper and combines them together with a neural network model.

3 Methodology

The word segmentation task is usually solved by character-level sequence labeling algorithm. Specifically, given a character sequence x, our model generates a

corresponding y, where y belongs to the collection of {'B', 'I', 'E', 'S'}. 'B' denotes the beginning position, 'I' denotes the middle position, 'E' denotes the ending positions of a word and 'S' denotes this position is a word of a single character. We use y to segment the sequence.

Table 1 shows an example of word segmentation on a Chinese sentence "中国外长将访问加拿大/The Chinese Foreign Minister will visit Canada".

中	国	外	长	将	访	问	加	拿	大
B	E	B	E	S	B	E	B	I	E

Table 1. An example of word segmentation on a Chinese sentence.

In this section, we first describe the features we utilize in this work and give the proposed feature-rich LSTM-based model.

3.1 Pretrained Character-Bigram Embedding

Previous works show that using pre-trained word embeddings helps the model to converge to better results compared to randomly initialized word embeddings in many NLP tasks. Similarly, we use pre-trained character-bigram embeddings instead of randomly initialized ones. An intuitive explanation is the pre-trained character-bigram embeddings carry more semantic information due to they are obtained on a large corpus. To the best of our knowledge, there hasn't been any work that uses pre-trained character-bigram embedding on word segmentation task.

The character-unigram embeddings we utilize are initialized randomly. To obtain the pre-trained character-bigram embeddings, we first convert the original character sequence to a bigram sequence. For example, the bigram sequence of sentence "我是中国人。" will be "我是 是中 中国 国人 人。". Then we can train bigram embeddings readily using word2vec[8] toolkit on the resulting bigram sequences.

3.2 Statistics-Based Features

Statistics-based features have been shown helpful for word segmentation task [10]. We'll introduce three kinds of statistics-based features we utilize in this part, including pointwise mutual information, accessor variety and punctuation variety.

We scale all the raw scores of statistics-based features with their z-scores, the z-score of raw score x is $\frac{x-\mu}{\sigma}$, where μ and σ are the mean and standard deviation of the raw score distribution, respectively. z-score measures the distance between the raw score and the population mean in the units of standard deviation. A z-score reflects the position of the original value in all values. It is a linear transformation of the original score and does not change the distribution of the original score.

Pointwise Mutual Information PMI (pointwise mutual information) is very helpful for word segmentation because word boundaries are more likely to occur between two characters with low PMI than high PMI. It has the ability to measure the closeness between characters. It has been used on word segmentation task in previous works.

The PMI values are computed through:

$$\texttt{PMI}\left(c_1, c_2\right) = \log \frac{P\left(c_1 c_2\right)}{P\left(c_1\right) P\left(c_2\right)} \tag{1}$$

where $P(c_1)$, $P(c_2)$ and $P(c_1c_2)$ are counted on the big corpus of raw data. $P(s)$ denotes the probability string s appears in the raw data.

Table 2 shows a snapshot of z-scored PMI, we find that two characters with high PMI tend to belong to the same word, Otherwise they tend to belong to different words.

Character Pair	PMI	Is a word?
中国 (China)	1.8448	yes
豚鼠 (Guinea pig)	2.9991	yes
我病 (I sick)	-0.9099	no
你去 (You go)	0.9693	no

Table 2. A snapshot of z-scored point mutual information.

Accessor Variety Accessor variety evaluates how independent a string is used, if a string is surrounded by a variety of different characters, it is very likely to be a word. This idea is introduced for identifying meaningful Chinese words in [2]. Given a string s, which consist of $l(2 \leq l \leq 3)$ characters, the *left accessor variety* $L_{av}^{l}(s)$ is defined as the number of distinct characters that precede s in a corpus. Similarly, the *right accessor variety* $R_{av}^{l}(s)$ is defined as the number of distinct characters that succeed s.

We obtained the accessor variety values from the large corpus of unlabeled data, and replaced their original values with their z-scored values, Table 3 shows a snapshot of z-scored accessor variety, the string with larger accessor variety value has more probability of being a word. Since the values are too large, they cannot be utilized by neural network models. We normalize them to [-1, 1] before using. The accessor variety we input at position i is shown as follows:

- Accessor variety of strings with length 2:
  $L_{av}^{2}(c_{[i:i+1]})$,$L_{av}^{2}(c_{[i+1:i+2]})$,$R_{av}^{2}(c_{[i-1:i]})$,$R_{av}^{2}(c_{[i-2:i-1]})$;
- Accessor variety of strings with length 3:
  $L_{av}^{3}(c_{[i:i+2]})$,$L_{av}^{3}(c_{[i+1:i+3]})$,$R_{av}^{3}(c_{[i-2:i]})$,$R_{av}^{3}(c_{[i-3:i-1]})$;

Character String	$L^l_{av}(s)$	$R^l_{av}(s)$	Is a word?
中国 (China)	36.8096	46.8093	yes
我们 (We)	27.4731	34.6770	yes
我病 (I sick)	0.5088	-0.0234	no
悄悄话 (Whispering)	0.9689	0.4563	yes

Table 3. A snapshot of z-scored accessor variety.

Punctuation Variety Punctuation variety is used to measure how often a string appears next to a punctuation mark, since punctuation marks are symbols that indicate the structure and organization of written language, if a string always appears next to the punctuation, it has more possibility of being a word. The definition of punctuation variety is very similar to the accessor variety. As defined by the previous work of [10], the punctuation variety $L^l_{pv}(s)$ if defined as the number of punctuation marks that precede string s. Similarly, $R^l_{pv}(s)$ is defined as the number of punctuation marks that succeed string s.

Table 4 shows a snapshot of z-scored punctuation variety obtained from large corpus of unlabeled data, the string with larger punctuation variety value has more probability of being a word. Since the values of punctuation variety are also too large for neural network models, we normalize them to [-1, 1] before using. We have two different kinds of features which can be the input feature of position i:

- Punctuation variety of strings with length 2: $L^2_{pv}(c_{[i:i+1]})$,$R^2_{pv}(c_{[i-1:i]})$;
- Punctuation variety of strings with length 3: $L^3_{pv}(c_{[i:i+2]})$,$R^3_{pv}(c_{[i-2:i]})$;

Character String	$L^l_{pv}(s)$	$R^l_{pv}(s)$	Is a word?
中国 (China)	240.9068	61.6721	yes
我们 (We)	92.0613	1.4849	yes
我病 (I sick)	-0.0286	-0.0452	no
悄悄话 (Whispering)	0.1021	0.1086	yes

Table 4. A snapshot of z-scored punctuation variety.

3.3 LSTM-based Model

Our model is based on bidirectional LSTM networks [3], which is very popular for sequence labeling tasks. A basic idea is feeding the character-unigram embedding to LSTM-based model and get the predicted label at position t using the corresponding hidden state h_t of bidirectional LSTM.

However, to most languages, a single character may not carry sufficient semantic information, so we decide to add character-bigram embedding into our model. And because of the size limitation of the labeled data, external information may be very useful to our model, we decide to add some statistics-based features which we have discussed in the previous part of this section to get our proposed feature-rich LSTM-based model.

Our input unit representation is calculated by concatenating character-unigram embedding, character-bigram embedding and all numerical features together, and pass it through a non-linear neural network layer, which can be represented as follows:

$$x_t = max\left\{0, W[B_{t-1}, B_t, U_t, \texttt{PMI}(c_{t-1}, c_t), \texttt{PMI}(c_t, c_{t+1}), ...] + b\right\} \tag{2}$$

where B_t denotes character-bigram embedding at position t, U_t denotes character-unigram embedding at position t, PMI denotes point mutual information between characters and other numerical features are omitted here.

Finally, we calculate the probability of label y_i at position t by the following equation:

$$p\left(y_i|h_t\right) = \frac{exp\left(g_i^T h_t + q_i\right)}{\sum_j exp\left(g_j^T h_t + q_j\right)} \tag{3}$$

where h_t is hidden state of bidirectional LSTM at position t, g_i is a column vector representing the embedding of the label i and q_i is a bias term for label i.

The architecture of our bidirectional LSTM-based model is illustrated in Figure 1.

4 Experiments

4.1 Data and Settings

We conduct experiments on three languages on CoNLL-2017 shared task [1] including traditional Chinese (zhT), Vietnamese (vi) and Japanese (ja). And we do further experiments on three simplified Chinese datasets: PKU and MSR from 2^{nd} SIGHAN backoff and Chinese Treebank 6.0 (CTB6.0). For the PKU and MSR datasets, last 10% of the training data are used as development data as [9] does. For CTB6.0 data, recommended data split is used. We covert all the double byte digits and letters into a single byte and then convert all continuous digits into one token '<Digit>' as our preprocess. The performance of our word segmentation model is evaluated by F-score. The statistics-based features and pre-trained character-bigram embeddings of three datasets from CoNLL-2017 shared task and three datasets of simplified Chinese are obtained from the raw data of Wikipedia provided by CoNLL-2017 shared task and the Chinese Gigawords, respectively. We use both character-unigram embedding and character-bigram embedding of 100 dimensional. The input dimension and hidden dimension of LSTM networks is set to 100 and 128, respectively. Table 5 shows the number of instances in each dataset.

[1] The data could be downloaded at http://universaldependencies.org/conll17/data.html

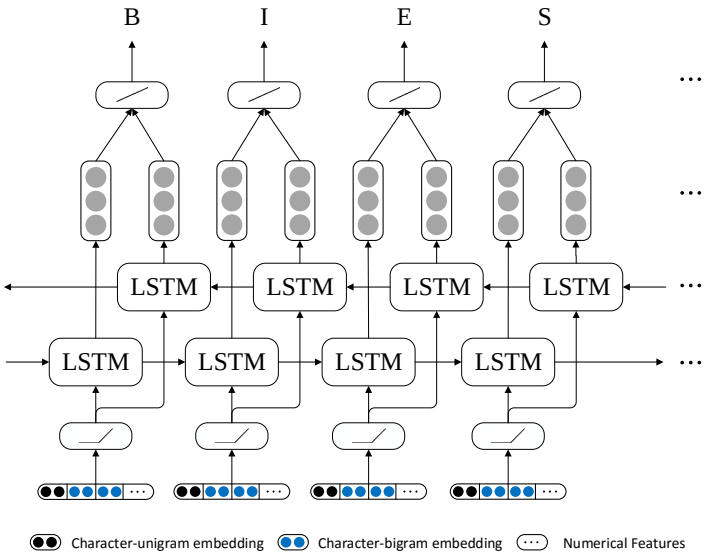

Fig. 1. An illustration of the LSTM-based model. The concatenated character-unigram embedding, two character-bigram embeddings and other numerical features are used as the input of the neural network after passed through an non-linear neural network layer.

Data set	CoNLL-2017			Simplified Chinese		
	zhT	vi	ja	CTB6.0	PKU	MSR
Training	3,997	1,400	7,164	23,416	17,149	78,232
Devel.	500	800	511	2,077	1,905	8,692
Test	500	800	557	2,796	1,944	3,985

Table 5. The number of instances in training, development and test data of 6 different datasets.

Our numerical features used in our experiments are all z-scored, including PMI, accessor variety and punctuation variety.

4.2 Experimental Results

On the following tables, 'Pre' denotes using pre-trained character-bigram embedding, 'PMI' denotes using z-scored point mutual information, 'AV' denotes using z-scored accessor variety and 'PV' denotes using z-scored punctuation variety.

Results on Development Dataset Table 6 shows the results on development dataset, evaluated by F-score. From Table 6, we can see numerical features

obtained on unlabeled data have improvement on small dataset (e.g. zhT and vi from CoNLL-2017 shared task) and have less effect on dataset with enough training data. All of point mutual information, accessor variety and punctuation variety are able to improve the performance of our model in a different level. We can also see, PMI is the most helpful feature especially on smaller dataset while the other two features get smaller improvement.

Data set	CoNLL-2017			Simplified Chinese		
	zhT	vi	ja	CTB6.0	PKU	MSR
	(3,997)	(1,400)	(7,164)	(23,416)	(17,149)	(78,232)
Baseline	92.00	89.29	93.70	95.09	96.04	97.06
Pre	94.27	91.81	94.28	95.53	96.46	97.13
Pre+PMI	95.22	93.07	94.83	95.56	96.57	97.30
Pre+PMI+AV	**95.77**	**93.57**	94.72	95.58	96.64	97.49
Pre+PMI+PV	95.56	93.31	94.81	**95.69**	96.65	**97.50**
Pre+PMI+AV+PV	95.47	93.34	**95.13**	95.62	**96.67**	97.39

Table 6. Results on development dataset, using different sets of features, evaluated by F-score. The numbers in parentheses denote the number of instances in corresponding training dataset.

We evaluate the Out-of-vocabulary (OOV) recall rate on the development dataset. As shown in Table 7, adding numerical statistics-based features significantly increases the OOV recall rate regardless of the size of the training dataset. The improvement is more obvious when the size of training dataset is smaller. All of pre-trained character-bigram embeddings, pointwise mutual information, accessor variety and punctuation variety have the ability to help the model to recognize more OOV words. For dataset which has high OOV rate, introducing more statistics-based features also gives more information which the model cannot learn in the training dataset. Increasing OOV recall rate directly improves the performance of the model.

Results on Evaluation Dataset At last, we compare our LSTM-based model with other state-of-the-art models in Table 8. The first block of Table 8 shows the non-neural CWS models and the second block shows the neural models. Both [7] and [14] used word-level information which we didn't use in this work. [13] use auto-segmented result of large scale unlabeled data. Our work tries to feed numerical statistics-based features to LSTM networks and gets competitive results. From Table 8 we can see that our model achieves the state-of-the-art performance on traditional Chinese and Vietnamese word segmentation dataset of CoNLL-2017 shared task, which has less training instances compared with other datasets. And on the other four datasets, we have comparable results to state-of-the-art performance.

Data set	CoNLL-2017			Simplified Chinese		
	zhT	vi	ja	CTB6.0	PKU	MSR
	(12.1)	(14.7)	(8.5)	(5.4)	(3.8)	(2.7)
Baseline	63.9	63.4	67.2	75.4	67.6	60.9
Pre	79.4	71.9	73.2	76.0	69.8	62.9
Pre+PMI	82.9	75.1	73.8	77.6	68.2	71.6
Pre+PMI+AV	**85.1**	**78.1**	72.7	75.5	71.5	71.1
Pre+PMI+PV	83.6	76.5	74.3	**78.2**	**72.5**	**71.7**
Pre+PMI+AV+PV	83.2	76.3	**75.3**	76.6	71.5	70.1

Table 7. OOV recall rate on development dataset, using different sets of features. The numbers in parentheses denote the OOV rate of corresponding development dataset.

5 Conclusion and Future Work

In this paper, we scale the value of statistic-based features in previous works with their z-scored values and feed the new values into our LSTM-based model as numerical features. Experiments show that it significantly improves the F-score on smaller datasets and it can also slightly enhance the performance on larger datasets. Also, this method shows greater generalization than those methods without statistic-based features counted on unlabeled data. We analyze the effect of statistic-based features by giving the OOV recall rate of each development dataset with different sets of features. Our model achieves state-of-the-art performance on two datasets of CoNLL-2017 shared task. And we have comparable results to state-of-the-art performance on the other datasets.

We plan to add more effective numerical features into neural network model and to try some new methods other than z-score method to get the numerical features.

Acknowledgements

We thank the anonymous reviewers for their valuable suggestions. This work was supported by the National Key Basic Research Program of China via grant 2014CB340503 and the National Natural Science Foundation of China (NSFC) via grant 61370164 and 61632011.

References

1. Cai, D., Zhao, H.: Neural word segmentation learning for chinese. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 409–420. Association for Computational Linguistics, Berlin, Germany (August 2016)
2. Feng, H., Chen, K., Deng, X., Zheng, W.: Accessor variety criteria for chinese word extraction. Computational Linguistics 30(1), 75–93 (2004)

genre	model	CoNLL-2017			Simplified Chinese		
		zhT	vi	ja	CTB6.0	PKU	MSR
non-NN	[Tseng, 2005] [12]	-	-	-	-	95.0	96.4
	[Zhang and Clark, 2007] [15]	-	-	-	-	95.1	97.2
	[Sun *et al.*, 2009] [11]	-	-	-	-	95.2	97.3
	[Wang *et al.*, 2011] [13]	-	-	-	**95.7**	-	-
NN	[Zheng *et al.*, 2013] [16]	-	-	-	-	92.4	93.3
	[Pei *et al.*, 2014] [9]	-	-	-	-	94.0	94.9
	[Pei *et al.*, 2014] w/bigram [9]	-	-	-	-	95.2	97.2
	[Kong *et al.*, 2015] [4]	-	-	-	-	90.6	90.7
	[Liu *et al.*, 2016] [7]	-	-	-	95.48	95.67	97.58
	[Zhang *et al.*, 2016] [14]	-	-	-	-	**95.7**	**97.7**
	[Cai *et al.*, 2016] [1]	-	-	-	-	95.5	96.5
	1^{st} on Offical Evaluation	94.57	87.30	**98.59***	-	-	-
	our best	**95.49**	**91.96**	95.09	95.26	95.44	97.32

Table 8. Comparison with the state-of-the-art word segmentation systems, evaluated by F-score. We still don't know the methods of 1^{st} systems of CoNLL-2017 shared task official evaluation. The 1^{st} system on Japanese of CoNLL-2017 shared task is an in-house system.

3. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
4. Kong, L., Dyer, C., Smith, N.A.: Segmental recurrent neural networks. arXiv preprint arXiv:1511.06018 (2015)
5. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML. vol. 1, pp. 282–289 (2001)
6. Liang, P.: Semi-supervised learning for natural language. Ph.D. thesis, Massachusetts Institute of Technology (2005)
7. Liu, Y., Che, W., Guo, J., Qin, B., Liu, T.: Exploring segment representations for neural segmentation models pp. 2880–2886 (2016)
8. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. International Conference on Learning Representations (ICLR) Workshop (2013)
9. Pei, W., Ge, T., Chang, B.: Max-margin tensor neural network for chinese word segmentation. In: ACL (1). pp. 293–303 (2014)
10. Sun, W., Xu, J.: Enhancing chinese word segmentation using unlabeled data. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 970–979. Association for Computational Linguistics (2011)
11. Sun, X., Zhang, Y., Matsuzaki, T., Tsuruoka, Y., Tsujii, J.: A discriminative latent variable chinese segmenter with hybrid word/character information. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 56–64. Association for Computational Linguistics (2009)
12. Tseng, H., Chang, P., Andrew, G., Jurafsky, D., Manning, C.: A conditional random field word segmenter for sighan bakeoff 2005. In: Proceedings of the fourth SIGHAN workshop on Chinese language Processing. vol. 171. Citeseer (2005)

13. Wang, Y., Jun'ichi Kazama, Y.T., Tsuruoka, Y., Chen, W., Zhang, Y., Torisawa, K.: Improving chinese word segmentation and pos tagging with semi-supervised methods using large auto-analyzed data. In: IJCNLP. pp. 309–317 (2011)
14. Zhang, M., Zhang, Y., Fu, G.: Transition-based neural word segmentation. Proceedings of the 54nd ACL (2016)
15. Zhang, Y., Clark, S.: Chinese segmentation with a word-based perceptron algorithm. In: Annual Meeting-Association for Computational Linguistics. vol. 45, p. 840 (2007)
16. Zheng, X., Chen, H., Xu, T.: Deep learning for chinese word segmentation and pos tagging. In: EMNLP. pp. 647–657 (2013)