A Chinese word segment model for energy literature based on Neural Networks with Electricity User Dictionary

Bochuan Song
Artificial Intelligence on Electric Power System
State Grid Corporation Joint Laboratory
Global Energy Interconnection Research Institute co.Ltd
Beijing, China
songbochuan@geiri.sgcc.com.cn

Bo Chai
Artificial Intelligence on Electric Power System
State Grid Corporation Joint Laboratory
Global Energy Interconnection Research Institute co.Ltd
Beijing, China
chaibo@geiri.sgcc.com.cn

Qiang Zhang
Artificial Intelligence on Electric Power System
State Grid Corporation Joint Laboratory
Global Energy Interconnection Research Institute co.Ltd
Beijing, China
zhangqiang1@geiri.sgcc.com.cn

Quanye Jia
Artificial Intelligence on Electric Power System
State Grid Corporation Joint Laboratory
Global Energy Interconnection Research Institute co.Ltd
Beijing, China
jiaquanye@geiri.sgcc.com.cn

***Abstract*—Traditional Chinese word segmentation (CWS) methods are based on supervised machine learning such as Condtional Random Fields(CRFs), Maximum Entropy(ME), whose features are mostly manual features. These manual features are often derived from local contexts. Currently, most state-of-art methods for Chinese word segmentation are based on neural networks. However these neural networks rarely introduct the user dictionary. We propose a LSTM-based Chinese word segmentation which can take advantage of the user dictionary. The experiments show that our model performs better than a popular segment tool in electricity domain. It is noticed that it achieves a better performance when transfered to a new domain using the user dictionary.**

***Keywords*-Chinese word segmentation, electricity user dictionary, neural networks**

I. Introduction

Words are the basic units to process for most Natural Language Processing (NLP) tasks. However, most east Asian languages are written without explicit word delimiters, including Chinese. As a result, word segmentation is usually the first step of NLP in east Asian languages process.

There are two obstacles for Chinese word segmentation. One is overlapping ambiguity. For example, the sentence "南京市长江大桥顺利通车"(Nanjing Yangtze River Bridge was successfully opened to traffic). The segmentation of the utterance can be "南京市(Nanjing City) / 长江大桥(Changjiang Bridge) / 顺利(successfully) / 通车(opened to traffic)". It can also be segmented as "南京(Nanjing City) / 市长(mayor) / 江大桥(Jiang Daqiao) / 顺利(successfully) / 通车(opened to traffic)". The middle character of an overlapping ambiguous string can constitute words with the characters to both their left and their right [1].

The other one is the out of vocabulary problem (OOV). For example, while "国家电网"(State Grid) appears as one word at test time, a model trained on a dataset where the character subsequence is segmented as two words "国家"(Country) and "电网"(Grid) will split the word "国家电网"(State Grid) into two words.. Some of the errors will be almost impossible to solve [2].

There have been a wide range of studies on Chinese word segmentation. Most methods formalize this task as a sequence labeling problem [2], [3]. In a supervised machine learning fashion, this problem may adopt various models e.g., Maximum Etropy [4] and Conditional Random Fields [5]. These methods usually heavily depend on manual features. Tseng et al. [6] designed a CRF segmenter using n-gram character features. There are more complicated features, e.g., accessor varitey criteria [7], conditional entropy features [8]. As neural networks can extract features on their own, neural models have been widely used for NLP tasks. For the task of CWS, Zheng et al. [3] adapted the general neural networks architecture for sequence labeling and used character embeddings as input to a two-layer network. Cai et al. [9] proposed a methods based on both character features and word embeddings. The beam search method is used to trade off the search complexity and search accuracy.

By introducing the user dictionary with the aim of solving the OOV problem, the traditional sequence labeling methods (e.g., CRFs) are limited by the context window to reach the long term history information. In addition, the neural network models cannot deal with the OOV problem, due to the lack of utilization of the user dictionary.

It is our goal to develop NLP technologies in the field of electricity. The Chinese word segmentation is the fundation of most NLP tasks, e.g., name entity recognition (NER) and information extraction (IE). A common model usually obtains a lower performance in a novel specific domain. It is necessary to study the CWS problem in the electricity field. Some prior work has been done to tackle the challenge brought by CWS in the area, e.g., collection of training data, and construction of an electricity user dictionary.

978-1-7281-5014-7/19/$31.00

In this paper, we proposed a method based on neural networks which can model all the previous history information rather than a fixed context window size as machine learning models do. In order to get better performance on out-of-domain corpus, the proposed model is designed to adopt the user dictionary. To the best of our knowledge, it is the first time to introduce user dictionary in neural networks to solve the CWS problem

The paper is organized as follows. Our methods are introduced in section II. The analysis and results are discussed in section III. Section IV presents the conclusion.

II. METHODS

Chinese word segmentation task is regarded as a character-based sequence labeling problem. Specifically, each character in a sentence is assigned with a label, i.e., $\mathcal{L} = \{B, M, E, S\}$, indicating the begin, middle, end of a word or a word with a single character. To segment a character sequence, we employ neural networks to extract the features and employ the CRFs to utilize the dependencies between tags. To improve the performance in a different domain, we introduce a electricity user dictionary which is implemented between the LSTM layer and the CRFs layer. Fig. 1 illustrates the proposed model.

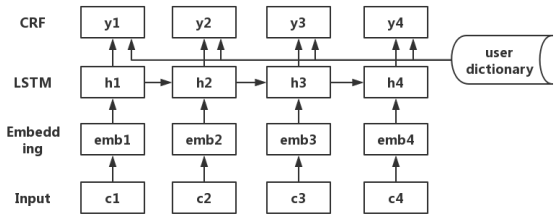

Figure 1. Our LSTM CRF User dict model.

A. Embedding

As neural network models are based on vector, firstly we generate the character embeddings randomly. Formally, we have a character dictionary $\mathcal{D}$ of size $|\mathcal{D}|$. Each character $\mathbf{c_i}$ is represented as a real-valued vector $\mathbf{c} \in R^{d_c}$, where d_c is the dimensionality of character vector. The character vectors are inserted into a matrix $\mathbf{M} \in R^{d_c \times |\mathcal{D}|}$. The embedding layer retrieves the character $\mathbf{c_i}$ according to its index to get its embedding.

B. LSTM

The LSTM neural network [10] is an extension of Recurrent Neural Networks(RNNs), which has been widely adopted in NLP tasks. RNNs are a family of neural networks that operate on sequential data. In the RNN, a sequence of vector $(\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_n})$ is taken as input, and another sequence of vector $(\boldsymbol{h_1}, \boldsymbol{h_2}, ..., \boldsymbol{h_n})$ is finally obtained. Although RNNs can learn long-dependencies in theory, but they usually failed in practical because of gradient explosing and vanishing problems [11]. The LSTM is designed to cope with the issues by introducing a memory-cell and three gate functions, i.e., input gate, forget gate and output gate. LSTMs have been shown to capture long-dependencies better than RNNs [12]. There also are variant implementation of LSTMs. In our model, we use the following implementation:

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{1}$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{3}$$
$$\tilde{c_t} = \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \tag{4}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c_t} \tag{5}$$
$$h_t = o_t \odot \sigma_h(c_t) \tag{6}$$

f_t, i_t, o_t are forget gate, input gate and output gate, respectively. $\odot$ is elementwise multiplication. x_t is the input of LSTMs at time step t and h_t is the output of LSTMs at time step t.

C. CRF

A simple but effective model use the h_t as feature to make independent classification by adding a softmax layer following the LSTM layer, which models the distribution as follows:

$$P(y_i) = P(y_i|x_1, x_2, ..., x_{i-1}) \tag{7}$$

However, it may fail when there are strong dependencies between tags. Chinese word segmentation is one such task, since the characterize interpretable sequences of tags imposes several hard constraints (e.g., tag I cannot follow tag E or tag S). So we use CRFs as labeling layer which can model these constraints. For an input sequence,

$$\boldsymbol{x} = (x_1, x_2, ..., x_n), \tag{8}$$

we consider $\mathbf{E}$ to be the matrix of scores derived from the output of the LSTM layer. $\mathbf{E}$ is of size $(n \times k)$, where k is the number of distinct tags. $E_{i,j}$ represents the score of the jth tag of the ith word in a sentence. For a sequence of predictions

$$\boldsymbol{y} = (y_1, y_2, ..., y_n), \tag{9}$$

we define its score to be

$$s(\mathbf{x}, \mathbf{y}) = \sum_i T_{y_{i-1}, y_i} + \sum_i E_{i, y_i} \tag{10}$$

in which $\mathbf{T}$ is the transition matrix of CRF, T_{y_{i-1}, y_i} means the score of a transition from tag y_{i-1} to tag y_i. A softmax over all tag sequences yield a probability of $\boldsymbol{y}$

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{e^{s(\boldsymbol{y}, \boldsymbol{x})}}{\sum_{\tilde{\boldsymbol{y}}} e^{\tilde{\boldsymbol{y}}, \boldsymbol{x}}} \tag{11}$$

For training, we maximum the $P(\boldsymbol{y}|\boldsymbol{x})$ where $\boldsymbol{y}$ is the truth ground tag sequence. During decoding, we predict the tag sequence that obtains the maximum score given by:

$$\hat{\boldsymbol{y}} = \underset{\hat{\boldsymbol{y}}}{\operatorname{argmin}}\ s(\hat{\boldsymbol{y}}, \boldsymbol{x}) \tag{12}$$

In evaluation stage, the Viterbi algorithm is utilized for decoding. The Viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states [13]. Its algorithm is described in Algorithm 1.

Algorithm 1: Viterbi Algorithm

Data: emission matrix, output of LSTM, $\mathbf{E}$, and transition matrix $\mathbf{T}$
Result: best path P

```
/* initialize the parameters.                    */
paths ← List;
scores ← List;
scores[0] ← E[start_tag, :];
for i ← 1 to seqlen do
    potentials ← scores[i − 1] + T;
    score, path ← max(potentials);
    paths[i] ← path;
    scores[i] ← score + E[i, :];
end
/* construct the most likely
   sequence backwards                            */
viterbi_score, viterbi_path ←
max(scores[seqlen, :]);
viterbi_paths[seqlen] ← [viterbi_path];
for i ← seqlen − 1 to 1 do
    viterbi_path ← paths[i − 1];
    viterbi_paths[i] ← viterbi_path;
end
```

D. User Dictionary

In traditional segmentation fashion, the user dictionary is a widely used method, e.g., n-gram, forward maximum matching. The mentioned models can achieve better performance with a well-organized user dictionary when applied in a novel-specific domain. As far as we know, there are not any neural-based models that adopt user dictionary. The user dictionary is used to combine the LSTM and the CRF to tackle the word segmentation problem. Firstly, we find all candidate words in a input sentence by looking up the user dictionary. Then, for each character in every candidate words, we add a weight to the corresponding E_{i,y_i} according to the charater position in the candidate word to get $\hat{E}_{i,y_i}$. Finally replace E_{i,y_i} by $\hat{E}_{i,y_i}$ in (10)

III. Experiment

A. Datasets and Metrics

We use two datasets, the PKU [14] and the Energy. The overview of the two datasets is presented in Table I.

Table I
Overview of Datasets

Dataset	Trainset	Devset	Testset
PKU	31479	4526	9078
Energy	9575	1137	2240

PKU dataset is collected from People Daily. It has been widely used for CWS task [8], [10], [15]. Energy dataset contains around 12,000 sentences. We also make a user dictionary, which contains about 9,100 words and is collected from other corpora. For evaluation, we use F1-measure. The formula is given as following:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{13}$$

$$Precision = \frac{N\_Right}{N\_Pred} \tag{14}$$

$$Recall = \frac{N\_Right}{N\_Gold} \tag{15}$$

The N_Right is the total number of correct predicted segment words. The N_Pred is the total number of predicted segment words. The N_Gold is the total number of ground truth words.

B. Hyper-parameters

Hyper-parameters of neural networks models has a strong impact on their performance. In our model, the size of character embeddings is 128. The hidden size of LSTM is 256. We use adam optimizer, and the learning rate is 0.01.

C. Results and Analysis

Serveral experiments have been conducted. Firstly, we compare our model in the general domain with the popular segment tool Jieba segmenter on PKU dataset. The results are showed in Table II.

Table II
Results on PKU

Model	Precision	Recall	F1
Jieba	0.8174	0.7829	0.7998
Our model	0.8932	0.9011	0.8971

It shows that the our model is obviously better than the widely used Jieba segmenter in the general domain.

Then we train our model on Energy trainset, and test both our model and Jieba segmenter on the testset of Energy. We also find that if we combine the trainsets of PKU and Energy our model will perform marginally better. The results are showed in Table III.

Table III
Results on Energy

Model	Trainset	Precision	Recall	F1
Jieba	-	0.7547	0.8157	0.7840
Jieba-user_dict	-	0.7720	0.8054	0.7884
Our model	Energy	0.7794	0.8387	0.8079
Our model	Energy+PKU	0.7875	0.8371	0.8116

As the Jieba segmenter gains a little inprovement on F1-measure score after the user dictionary employed (the comparison between Jieba and Jieba-user_dict in Table III), it shows that the dictionary introduce some domain information. As a result, the comparison between

our model trained on Energy and Jieba-user_dict shows that our model also gains inprovement with Jieba segmenter. It is noticed that our model perform better after we combine trainsets of PKU and Energy. The reason can be illustrated as that Energy dataset is not sufficient. The bigger datasets, PKU, can introduce some common knowledge into our model, which has positive effect on Chinese word segmentation. It is a simple data transfer mode of transfer learning.

Finally we conduct experiment to evaluate the effects of the user dictionay on our neural-based model. We have two sets of models. The first one is trained on Energy trainset, and is tested on Energy testset. The second one is trained on PKU trainset, and is tested on Energy testset. The results are showed in Table IV.

Table IV
RESULTS ON USER DICTIONARY NEURAL MODEL

Model	Trainset	Precision	Recall	F1
our model	PKU	0.5770	0.6812	0.6248
our model user dict	PKU	0.6001	0.6762	0.6359
Our model	Energy	0.7794	0.8387	0.8079
our model user dict	Energy	0.7834	0.8180	0.8003

In order to make the difference clear, we add our model without user dictionary which trained on Energy into the Table IV. The first two lines of Table IV shows that the user dictionary has positive effect on segmentation, since the model with the user dictionary gains 1.11% inprovement on F1-measure score compared with model without the user dictionary. These two model are trained on PKU dataset. it is noteworthy that when the trainset changed from PKU to Energy, the results are different. In the last two line of Table IV, it shows that the introducing of the user dictionary degrades the performance of the proposed model when trained on Energy.

We investigate the overlapped words among the dictionary and testset of Energy. In the testset, there are 919 sentences where there are subsequences of characters, which can compose words in the dictionary. However among these sentences, only 450 sentences have the correct words after segment. When the our model is trained on Energy, it learns the sufficient knowledge of the electricity domain. It is the gap between the domain from which the dictionary collected and the Energy dataset that degrades the performance when our model is trained on Energy with the user dictionary. For a example, there is a word "能源基地"(Energy base) in the dictionary. For a sentence in the testset, "新形势下能源基地开发潜力..."(Energy base development potential under the new situation), the correct segmentation is "新(new) / 形势(situation) / 下(under) / 能源(energy) / 基地(base) / 开发(development) / 潜力(potential)". But the word in the dictionary will force the character sequence "能源基地"(energy base) to be one word. When the model is trained on general domain dataset, PKU, it lack of the domain knowledge. Thus the user dictionary is supposed to improve the performance on Energy testset.

IV. CONCLUSION

We proposed a neural-based model with the user dictionary. Experiments show that the proposed model gains improvement on both general domain and energy domain compared with Jieba segmenter. One attractive feature of the proposed model is that it can adopt a user dictionary, which can transfer the proposed model to a new domain with little work. The experiments show that the proposed model with a user dictionary obtains a better performance when trained on a general domain and tested on a new domain compared to the model without a user dictionary.

In the future, we plan to introduce the user dictionary into the transition matrix of the CRF, which will make the introducing more seamlessly and more flexibly. In addition, our basic model, LSTM-CRF, still need to be improved. Some word-based segmentation mechanism can be introduced in the future work.

ACKNOWLEDGMENT

This research was supported by State Grid Company Research Project "Task-driven multi-round dialogue generation based on semantic understanding" under grant 5455HJ190008.

REFERENCES

[1] Ma, Guojie, Xingshan Li, and Keith Rayner. "Word segmentation of overlapping ambiguous strings during Chinese reading." Journal of Experimental Psychology: Human Perception and Performance 40.3 (2014): 1046.

[2] Ma, Ji, Kuzman Ganchev, and David Weiss. "State-of-the-art Chinese word segmentation with bi-lstms." arXiv preprint arXiv:1808.06511 (2018).

[3] Zheng, Xiaoqing, Hanyang Chen, and Tianyu Xu. "Deep learning for Chinese word segmentation and POS tagging." Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. 2013.

[4] Low, Jin Kiat, Hwee Tou Ng, and Wenyuan Guo. "A maximum entropy approach to Chinese word segmentation." Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing. 2005.

[5] Peng, Fuchun, Fangfang Feng, and Andrew McCallum. "Chinese segmentation and new word detection using conditional random fields." Proceedings of the 20th international conference on Computational Linguistics. Association for Computational Linguistics, 2004.

[6] Tseng, Huihsin, et al. "A conditional random field word segmenter for sighan bakeoff 2005." Proceedings of the fourth SIGHAN workshop on Chinese language Processing. 2005.

[7] Feng, Haodi, et al. "Accessor variety criteria for Chinese word extraction." Computational Linguistics 30.1 (2004): 75-93.

[8] Gao, Qin, and Vogel Stephan. "A multi-layer Chinese word segmentation system optimized for out-of-domain tasks." CIPS-SIGHAN Joint Conference on Chinese Language Processing. 2010.

[9] Cai, Deng, and Hai Zhao. “Neural word segmentation learning for Chinese.” arXiv preprint arXiv:1606.04300 (2016).

[10] Hochreiter, Sepp, and Jürgen Schmidhuber. “Long short-term memory.” Neural computation 9.8 (1997): 1735-1780.

[11] Hochreiter, Sepp, et al. “Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.” (2001).

[12] Hochreiter, Sepp, and Jürgen Schmidhuber. “Long short-term memory.” Neural computation 9.8 (1997): 1735-1780.

[13] Lou, H-L. “Implementing the Viterbi algorithm.” IEEE Signal processing magazine 12.5 (1995): 42-52.

[14] Emerson, Thomas. “The second international Chinese word segmentation bakeoff.” Proceedings of the fourth SIGHAN workshop on Chinese language Processing. 2005.

[15] Huang, Weipeng, et al. “Toward Fast and Accurate Neural Chinese Word Segmentation with Multi-Criteria Learning.” arXiv preprint arXiv:1903.04190 (2019).