Dependency-based Gated Recursive Neural Network for Chinese Word Segmentation

Jingjing Xu and **Xu Sun**
MOE Key Laboratory of Computational Linguistics, Peking University
School of Electronics Engineering and Computer Science, Peking University
{xujingjing, xusun}@pku.edu.cn

Abstract

Recently, many neural network models have been applied to Chinese word segmentation. However, such models focus more on collecting local information while long distance dependencies are not well learned. To integrate local features with long distance dependencies, we propose a dependency-based gated recursive neural network. Local features are first collected by bi-directional long short term memory network, then combined and refined to long distance dependencies via gated recursive neural network. Experimental results show that our model is a competitive model for Chinese word segmentation.

1 Introduction

Word segmentation is an important pre-process step in Chinese language processing. Most widely used approaches treat Chinese word segmentation (CWS) task as a sequence labeling problem in which each character in the input sequence is assigned with a tag. Many previous approaches have been effectively applied to CWS problem (Lafferty et al., 2001; Xue and Shen, 2003; Sun et al., 2012; Sun, 2014; Sun et al., 2013; Cheng et al., 2015). However, these approaches incorporated many handcrafted features, thus restricting the generalization ability of these models. Neural network models have the advantage of minimizing the effort in feature engineering. Collobert et al. (2011) developed a general neural network architecture for sequence labeling tasks. Following this work, neural network approaches have been well studied and widely applied to CWS task with good results (Zheng et al., 2013; Pei et al., 2014; Ma and Hinrichs, 2015; Chen et al., 2015).

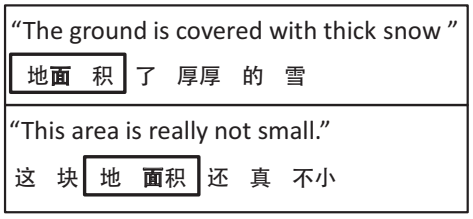


Figure 1: An illustration for the segmentation ambiguity. The character "面" is labeled as "E" (end of word) in the top sentence while labeled as "B" (begin of word) in the bottom one even though "面" has the same adjacent characters, "地" and "积".

However, these models focus more on collecting local features while long distance dependencies are not well learned. In fact, relying on the information of adjacent words is not enough for CWS task. An example is shown in Figure 1. The character "面" has different tags in two sentences, even with the same adjacent characters, " 地" and " 积". Only long distance dependencies can help the model recognize tag correctly in this example. Thus, long distance information is an important factor for CWS task.

The main limitation of chain structure for sequence labeling is that long distance dependencies decay inevitably. Though forget gate mechanism is added, it is difficult for bi-directional long short term memory network (Bi-LSTM), a kind of chain structure, to avoid this problem. In general, tree structure works better than chain structure to model long term information. Therefore, we use gated recursive neural network (GRNN) (Chen et al., 2015) which is a kind of tree structure to capture long distance dependencies.

Motivated by the fact, we propose the dependency-based gated recursive neural network (DGRNN) to integrate local features with long dis-

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 567–572,
Berlin, Germany, August 7-12, 2016.

tance dependencies. Figure 2 shows the structure of DGRNN. First of all, local features are collected by Bi-LSTM. Secondly, GRNN recursively combines and refines local features to capture long distance dependencies. Finally, with the help of local features and long distance dependencies, our model generates the probability of the tag of word.

The main contributions of the paper are as follows:

- We present the dependency-based gated recursive neural network to combine local features with long distance dependencies.
- To verify the effectiveness of the proposed approach, we conduct experiments on three widely used datasets. Our proposed model achieves the best performance compared with other state-of-the-art approaches.

2 Dependency-based Gated Recursive Neural Network

In order to capture local features and long distance dependencies, we propose dependency-based gated recursive neural network. Figure 2 illustrates the structure of the model.

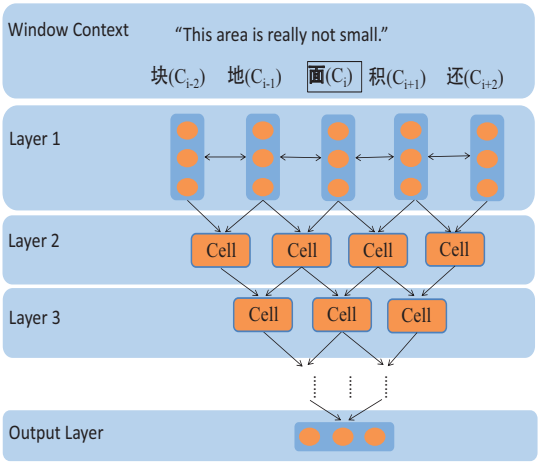


Figure 2: Architecture of DGRNN for Chinese Word Segmentation. Cell is the basic unit of GRNN.

2.1 Collect Local Features

We use bi-directional long short term memory (Bi-LSTM) with single layer to collect local features. Bi-LSTM is composed of two directional

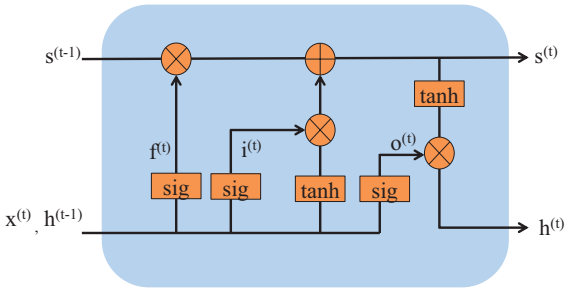


Figure 3: Structure of LSTM unit. The behavior of the LSTM cell is controlled by three "gates", namely input gate $i^{(t)}$, forget gate $f^{(t)}$ and output gate $o^{(t)}$.

long short term memory networks with single layer, which can model word representation with context information. Figure 3 shows the calculation process of LSTM. The behavior of LSTM cell is controlled by three "gates", namely input gate $i^{(t)}$, forget gate $f^{(t)}$ and output gate $o^{(t)}$. The input of LSTM cell are $x^{(t)}$, $s^{(t-1)}$ and $h^{(t-1)}$. $x^{(t)}$ is the character embeddings of input sentence. $s^{(t-1)}$ and $h^{(t-1)}$ stand for the state and output of the former LSTM cell, respectively. The core of the LSTM model is $s^{(t)}$, which is computed using the former state of cell and two gates, $i^{(t)}$ and $f^{(t)}$. In the end, the output of LSTM cell $h^{(t)}$ is calculated making use of $s^{(t)}$ and $o^{(t)}$.

2.2 Refine Long Distance Dependencies

GRNN recursively combines and refines local features to capture long distance dependencies. The structure of GRNN is like a binary tree, where every two continuous vectors in a sentence is combined to form a new vector. For a sequence s with length n, there are n layers in total. Figure 4 shows the calculation process of GRNN cell. The core of GRNN cell are two kinds of gates, reset gates, r_L, r_R, and update gates z. Reset gates control how to adjust the proportion of the input h_{i-1} and h_i, which results to the current new activation $h^{'}$. By the update gates, the activation of an output neuron can be regarded as a choice among the current new activation $h^{'}$, the left child h_{i-1} and the right child h_i.

2.3 Loss Function

Following the work of Pei et al. (2014), we adopt the max-margin criterion as loss function. For an input sentence $c_{[1:n]}$ with a tag sequence $t_{[1:n]}$, a sentence-level score is given by the sum of net-

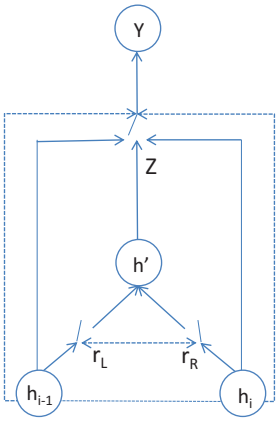


Figure 4: The structure of GRNN cell.

work scores:

$$s(c_{[1:n]}, t_{[1:n]}, \theta) = \sum_{i=1}^{n} f_\theta(t_i | c_{[i-2:i+2]}) \quad (1)$$

where $s(c_{[1:n]}, t_{[1:n]}, \theta)$ is the sentence-level score. n is the length of $c_{[1:n]}$. $f_\theta(t_i|c_{[i-2:i+2]})$ is the score output for tag t_i at the i_{th} character by the network with parameters θ.

We define a structured margin loss $\Delta(y_i, \widehat{y})$ for predicting a tag sequence $\widehat{y}$ and a given correct tag sequence y_i:

$$\Delta(y_i, \widehat{y}) = \sum_{j=1}^{n} \kappa \mathbf{1}\{y_{i,j} \neq y_i\} \quad (2)$$

where κ is a discount parameter. This leads to the regularized objective function for m training examples:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} l_i(\theta) + \frac{\lambda}{2} \|\theta\|^2 \quad (3)$$

$$\begin{aligned} l_i(\theta) \quad = \quad & \max_{\widehat{y} \subseteq Y(x_i)} ((s(x_i, \widehat{y}, \theta) \\ & + \Delta(y_i, \widehat{y})) - s(x_i, y_i, \theta)) \end{aligned} \quad (4)$$

where $J(\theta)$ is a loss function with parameters θ. λ is regularization factor. By minimizing this object, the score of the correct tag sequence y_i is increased and score of the highest scoring incorrect tag sequence $\widehat{y}$ is decreased.

2.4 Amplification Gate and Training

A direct adaptive method for faster backpropagation learning method (RPROP) (Riedmiller and Braun, 1993) was a practical adaptive learning method to train large neural networks. We use mini-batch version RPROP (RMSPROP) (Hinton, 2012) to minimize the loss function.

Intuitively, extra hidden layers are able to improve accuracy performance. However, it is common that extra hidden layers decrease classification accuracy. This is mainly because extra hidden layers lead to the inadequate training of later layers due to the vanishing gradient problem. This problem will decline the utilization of local and long distance information in our model. To overcome this problem, we propose a simple amplification gate mechanism which appropriately expands the value of gradient while not changing the direction.

Higher amplification may not always perform better while lower value may bring about the unsatisfied result. Therefore, the amplification gate must be carefully selected. Large magnification will cause expanding gradient problem. On the contrary, small amplification gate will hardly reach the desired effect. Thus, we introduce the threshold mechanism to guarantee the robustness of the algorithm, where gradient which is greater than threshold will not be expanded. Amplification gate of difference layer is distinct. For every sample, the training procedure is as follows.

First, recursively calculate m_t and v_t which depend on the gradient of time $t-1$ or the square of gradient respectively. β_1 and β_2 aim to control the impact of last state.

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \quad (5)$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \quad (6)$$

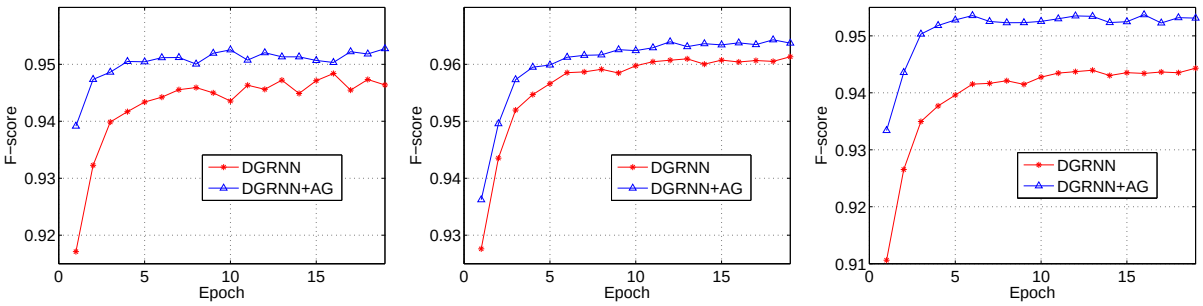
Second, calculate $\Delta W(t)$ based on v_t and square of m_t. ϵ and μ are smooth parameters.

$$M(w, t) = v_t - m_t^2 \quad (7)$$

$$\Delta W(t) = \frac{\epsilon g_{t,i}}{\sqrt{M(w,t)} + \mu} \quad (8)$$

Third, update weight based on the amplification gate and $\Delta W(t)$. The parameter update for the i_{th} parameter for the $\Theta_{t,i}$ at time step t with amplification gate γ is as follows:

$$\Theta_{t,i} = \Theta_{t,i} - \gamma \Delta W(t) \quad (9)$$



(a) PKU (b) MSRA (c) CTB6

Figure 5: Results for DGRNN with amplification gate (AG) on three development datasets.

3 Experiments

3.1 Data and Settings

We evaluate our proposed approach on three datasets, PKU, MSRA and CTB6. The PKU and MSRA data both are provided by the second International Chinese Word Segmentation Bakeoff (Emerson, 2005) and CTB6 is from Chinese TreeBank 6.0[1] (Xue et al., 2005). We randomly divide the whole training data into the 90% sentences as training set and the rest 10% sentences as development set. All datasets are preprocessed by replacing the Chinese idioms and the continuous English characters. The character embeddings are pre-trained on unlabeled data, Chinese Gigaword corpus[2]. We use MSRA dataset to preprocess model weights before training on CTB6 and PKU datasets.

Following previous work and our experimental results, hyper parameters configurations are set as follows: minibatch size $n = 16$, window size $w = 5$, character embedding size $d_1 = 100$, amplification gate range $\gamma = [0, 4]$ and margin loss discount $\kappa = 0.2$. All weight matrixes are diagonal matrixes and randomly initialized by normal distribution.

3.2 Experimental Results and Discussions

We first compare our model with baseline methods, Bi-LSTM and GRNN on three datasets. The results evaluated by F-score (F_1 score) are reported in Table 1.

- **Bi-LSTM**. First, the output of Bi-LSTM is concatenated to a vector. Second, softmax layer takes the vector as input and generates each tag probability.

[1]https://catalog.ldc.upenn.edu/LDC2007T36
[2]https://catalog.ldc.upenn.edu/LDC2003T09

Model (Unigram)	PKU	MSRA	CTB6
Bi-LSTM	95.0	95.8	95.2
GRNN	95.8	96.2	95.5
Pei et al. (2014)	94.0	94.9	*
Chen et al. (2015)	**96.1**	96.2	95.6
DGRNN	**96.1**	**96.3**	**95.8**

Table 1: Comparisons for DGRNN and other neural approaches based on traditional unigram embeddings.

Model	PKU	MSRA	CTB6
Zhang et al. (2006)	95.1	97.1	*
Zhang et al. (2007)	94.5	97.2	*
Sun et al. (2009)	95.2	97.3	*
Sun et al. (2012)	95.4	**97.4**	*
Zhang et al. (2013)	**96.1**	**97.4**	*
DGRNN	**96.1**	96.3	**95.8**

Table 2: Comparisons for DGRNN and state-of-the-art non-neural network approaches on F-score.

- **GRNN**. The structure of GRNN is recursive. GRNN combines adjacent word vectors to the more abstract representation in bottom-up way.

Furthermore, we conduct experiments with amplification gate on three development datasets. Figure 5 shows that amplification gate significantly increases F-score on three datasets. Amplification even achieves 0.9% improvement on CTB6 dataset. It is demonstrated that amplification gate is an effective mechanism.

We compare our proposed model with previous neural approaches on PKU, MSRA and CTB6 test datasets. Experimental results are reported in Table 1. It can be clearly seen that our approach achieves the best results compared with

Dataset	Model	Result
MSRA	Bi-LSTM	t = 5.94, $p < 1 \times 10^{-4}$
	GRNN	t = 1.22, $p = 0.22$
PKU	Bi-LSTM	t = 15.54, $p < 1 \times 10^{-4}$
	GRNN	t = 4.43, $p < 1 \times 10^{-4}$
CTB6	Bi-LSTM	t = 5.01, $p < 1 \times 10^{-4}$
	GRNN	t = 2.55, $p = 2.48 \times 10^{-2}$

Table 3: The t-test results for DGRNN and baselines.

other neural networks on traditional unigram embeddings. It is possible that bigram embeddings may achieve better results. With the help of bigram embeddings, Pei et al. (2014) can achieve 95.2% and 97.2% F-scores on PKU and MSRA datasets and Chen et al. (2015) can achieve 96.4%, 97.6% and 95.8% F-scores on PKU, MSRA and CTB6 datasets. However, performance varies among these bigram models since they have different ways of involving bigram embeddings. Besides, the training speed would be very slow after adding bigram embeddings. Therefore, we only compare our model on traditional unigram embeddings.

We also compare DGRNN with other state-of-the-art non-neural networks, as shown in Table 2. Chen et al. (2015) implements the work of Sun and Xu (2011) on CTB6 dataset and achieves 95.7% F-score. We achieve the best result on PKU dataset only with unigram embeddings. The experimental results show that our model is a competitive model for Chinese word segmentation.

3.3 Statistical Significance Tests

We use the t-test to intuitively show the improvement of DGRNN over baselines. According to the results shown in Table 3, we can draw a conclusion that, by conventional criteria, this improvement is considered to be statistically significant between DGRNN with baselines, except for GRNN approach on MSRA dataset.

4 Conclusions

In this work, we propose dependency-based recursive neural network to combine local features with long distance dependencies, which achieves substantial improvement over the state-of-the-art approaches. Our work indicates that long distance dependencies can improve the performance of local segmenter. In the future, we will study alternative ways of modeling long distance dependencies.

5 Acknowledgments

We thank Xiaoyan Cai for her valuable suggestions. This work was supported in part by National Natural Science Foundation of China (No. 61300063), National High Technology Research and Development Program of China (863 Program, No. 2015AA015404), and Doctoral Fund of Ministry of Education of China (No. 20130001120004). Xu Sun is the corresponding author.

References

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, and Xuanjing Huang. 2015. Gated recursive neural network for chinese word segmentation. In *ACL (1)*, pages 1744–1753. The Association for Computer Linguistics.

Fei Cheng, Kevin Duh, and Yuji Matsumoto. 2015. Synthetic word parsing improves chinese word segmentation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 262–267, Beijing, China, July. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November.

Thomas Emerson. 2005. The second international chinese word segmentation bakeoff. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, pages 123–133.

G. Hinton. 2012. Lecture 6.5: rmsprop: divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning.

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*, number 8 in ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Jianqiang Ma and Erhard Hinrichs. 2015. Accurate linear-time chinese word segmentation via embedding matching. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1733–1743, Beijing, China, July. Association for Computational Linguistics.

Wenzhe Pei, Tao Ge, and Baobao Chang. 2014. Max-margin tensor neural network for chinese word segmentation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 293–303, Baltimore, Maryland, June. Association for Computational Linguistics.

Martin Riedmiller and Heinrich Braun. 1993. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS*, pages 586–591.

Weiwei Sun and Jia Xu. 2011. Enhancing chinese word segmentation using unlabeled data. In *Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John Mcintyre Conference Centre, Edinburgh, Uk, A Meeting of Sigdat, A Special Interest Group of the ACL*, pages 970–979.

Xu Sun, Yaozhong Zhang, Takuya Matsuzaki, Yoshimasa Tsuruoka, and Jun'ichi Tsujii. 2009. 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*, pages 56–64, Boulder, Colorado, June. Association for Computational Linguistics.

Xu Sun, Houfeng Wang, and Wenjie Li. 2012. Fast online training with frequency-adaptive learning rates for chinese word segmentation and new word detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 253–262, Jeju Island, Korea, July. Association for Computational Linguistics.

Xu Sun, Yao zhong Zhang, Takuya Matsuzaki, Yoshimasa Tsuruoka, and Jun'ichi Tsujii. 2013. Probabilistic chinese word segmentation with non-local information and stochastic training. *Inf. Process. Manage.*, 49(3):626–636.

Xu Sun. 2014. Structure regularization for structured prediction. In *Advances in Neural Information Processing Systems 27*, pages 2402–2410.

N. Xue and L. Shen. 2003. Chinese Word Segmentation as LMR Tagging. In *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing*.

Naiwen Xue, Fei Xia, Fu-dong Chiou, and Marta Palmer. 2005. The Penn Chinese TreeBank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(2):207–238, June.

Yue Zhang and Stephen Clark. 2007. Chinese segmentation with a word-based perceptron algorithm. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 840–847, Prague, Czech Republic, June. Association for Computational Linguistics.

Ruiqiang Zhang, Genichiro Kikui, and Eiichiro Sumita. 2006. Subword-based tagging by conditional random fields for chinese word segmentation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, pages 193–196, Stroudsburg, PA, USA. Association for Computational Linguistics.

Longkai Zhang, Houfeng Wang, Xu Sun, and Mairgup Mansur. 2013. Exploring representations from unlabeled data with co-training for chinese word segmentation. In *EMNLP*, pages 311–321. ACL.

Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. 2013. Deep learning for chinese word segmentation and pos tagging. In *EMNLP*, pages 647–657. ACL.