# Neural Domain Adaptation with Contextualized Character Embedding for Chinese Word Segmentation

Zuyi Bao, Si Li, Sheng Gao, and Weiran Xu

Beijing University of Posts and Telecommunications, Beijing,
baozuyi@bupt.edu.cn, lisi@bupt.edu.cn,
gaosheng@bupt.edu.cn, xuweiran@bupt.edu.cn

**Abstract.** There has a large scale annotated newswire data for Chinese word segmentation. However, some research proves that the performance of the segmenter has significant decrease when applying the model trained on the newswire to other domain, such as patent and literature. The same character appeared in different words may be in different position and with different meaning. In this paper, we introduce contextualized character embedding to neural domain adaptation for Chinese word segmentation. The contextualized character embedding aims to capture the useful dimension in embedding for target domain. The experiment results show that the proposed method achieves competitive performance with previous Chinese word segmentation domain adaptation methods.

**Keywords:** Chinese word segmentation, contextualized character embedding, domain adaptation, neural network

## 1 Introduction

Chinese word segmentation is a necessary step for Chinese syntactic analysis due to Chinese text comes without word delimiters. Some state-of-the-art Chinese word segmentation systems with statistical techniques [15, 37, 24, 28, 16, 38, 26, 14, 34] reported high accuracy with large-scale annotated dataset, such as the Chinese TreeBank (CTB) [31], Peking University and Microsoft Research [12]. However, in actual use, the performance of the segmenter is not satisfying. As large-scale human annotated corpora mainly focus on domains like newswire, word segmentation systems trained on these corpora often suffer a rapid decrease in performance when they are used in other domains such as patents and literature [20, 18, 25]. In this paper, we consider such problem as *domain adaptation* [11] task.

Until now, two kinds of domain adaptation tasks are studied for Chinese word segmentation. One is annotation standard adaptation [15, 7], the other is document type adaptation [20, 21, 35, 25, 19]. The annotation standard adaptation aims to explore the common underlying knowledge from the corpora with different annotation standards. The document type adaptation is to use one domain document data to label the other domain document, such as using newswire document to label novel document. In this paper, we focus on the document type adaptation. Most previous research [20, 19] is based on the hand-crafted model which is difficult and time-consuming. In this paper, we adopt neural domain adaptation method for Chinese word segmentation.

In domain adaptation, the training data domain is often called *source* domain, and the testing data domain is often called *target* domain. Domain adaptation tries to resolve the problem that the training data and testing data are sampled from different distributions. The aim of domain adaptation is to learn a classifier which is trained on data mainly or all from *source* domain but generalizes well on *target* domain. In this paper, we focus on the semi-supervised domain adaptation [10] where annotated data is only available in *source* domain.

During Chinese word segmentation, the same character appeared in different words may be in different position and with different meaning. This observation is caused by the ambiguity of the character. The paper [29] indicates that much of ambiguity in word meaning can be resolved by considering surrounding words. This clue is also suitable for the character. The neural segmenter is usually based on character embeddings. In this paper, we follow this hypothesis that only a few dimensions in the source domain character embeddings are relevant to the target domain and we can turn off most of irrelevant dimensions. We introduce a mask network to turn off some dimensions of character embeddings by contextualizing a character embedding vector. Then the contextualized character embeddings are used in the neural segmenter. Our contributions are as follows:

(1) We introduce a mask model to adaptively mask out each dimension of the source and target embedding vectors to build the contextualized character embedding.

(2) We propose a neural domain adaptation segmenter with contextualized character embedding and show the effectiveness in the experiments.

## 2 Method

### 2.1 Contextualized Character Embedding

The amount of commonly used Chinese characters are limited, the meaning of each Chinese character is quite ambiguous. The same character appeared in different words may be in different position and with different meaning. Commonly used character embeddings do not distinguish each meaning of the character and are a mixture of every meaning. In this section, we introduce a contextualized character embedding for Chinese word segmentation.

Embedding is usually a $n$-dimension vector for each character. The $n$-dimension can be viewed as $n$ hidden semantics dimensions. We assume that one meaning of the character may be represented by some of $n$ hidden semantics dimensions. According to this assumption, the semantic mask is generated by the contextual information to specify the meaning of each character. Let $x_i$ refers to the embedding of $i$-th character, a sentence of $n$ characters can be represented as $x_1, x_2, \cdots, x_n$. As it is believed that the contextual information from a window size of $5$ characters may be sufficient for Chinese word segmentation [38], we employ a window size of $5$ characters to generate the semantic mask. Let $w_i = [x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}]$ refers to a concatenate of $5$ character embeddings around $i$-th character. The semantic mask $mask_i$ is generated as:

$$mask_i = \sigma(Ww_i + b), \tag{1}$$

where $\sigma$ is the sigmoid function, $W$ and $b$ are weight matrix and bias term. Then the contextualized character embedding $c_i$ is generated by masking the character embedding $x_i$ as:

$$c_i = mask_i \odot x_i, \tag{2}$$

where $\odot$ is element-wise product.

### 2.2 Character Sequence Auto-encoder

In order to train the contextualized character embedding $c_i$, we propose a unsupervised character sequence to sequence auto-encoder in this section. The architecture of our character sequence to sequence auto-encoder is similar to [27], but our auto-encoder is trained by rebuilding its input sequence. Let $c_i$ refers to the contextualized embedding of $i$-th character, a sentence of $n$ characters $y_1, y_2, \cdots, y_n$ can be represented as $c_1, c_2, \cdots, c_n$. An encoder is employed to map the sentence into a fix sized vector $h_n$. Then an decoder is employed to map the fix-sized vector $h_n$ to the original sentence again. Following the implement of [27], we use Long Short-Term Memory (LSTM) network to model the encoder and decoder. The description of a LSTM unit at time step $t$ is defined as follows:

$$i_t, f_t, o_t = \sigma(W_g c_t + U_g h_{t-1} + b_g), \tag{3}$$

$$\tilde{C}_t = \tanh(W_c c_t + U_c h_{t-1} + b_c), \tag{4}$$

$$C_t = i_t \odot \tilde{C}_t + f_t \odot C_{t-1}, \tag{5}$$

$$h_t = o_t \odot \tanh(C_t), \tag{6}$$

where $c_t$ is the input embeddings, $\sigma$, $\tanh$ are the sigmoid and hyperbolic tangent function, $\odot$ is element-wise product, $W$ and $U$ are weight matrices, $b$ is bias term. The $n$-th $h^{encoder}$ of encoder is used as fix-sized vector $h_n^{encoder}$. Then the initial hidden state $h_0^{decoder}$ is initialized as $h_n^{encoder}$. During the decoding, a special symbol *GO* is used as input and a hidden state $h^{decoder}$ is generated at every timestep. Then a softmax layer is appended for predicting the characters of the original sentence. The network is optimized by maximizing the likelihood:

$$\arg\max_{\theta} p(y_1^n | h_n^{encoder}) = \arg\max_{\theta} \prod_{t=1}^{n} p(y_t | h_{t-1}^{decoder}), \tag{7}$$

where $\theta$ is the weight arguments in the contextualized character embedding and character sequence to sequence auto-encoder, $y_t$ is the *t*-th character, the subsequence $y_1^n = (y_1, y_2, \cdots, y_{n-1}, y_n)$, the $h_0^{decoder}$ is initialized by the $h_n^{encoder}$.

### 2.3 Neural Segmenter

The neural segmenter is built above the contextualized character embedding. In this paper, we take convolutional neural segmenter as an example, but our method is not limited by the architecture of neural networks. The segmenter is simplified from [4], we

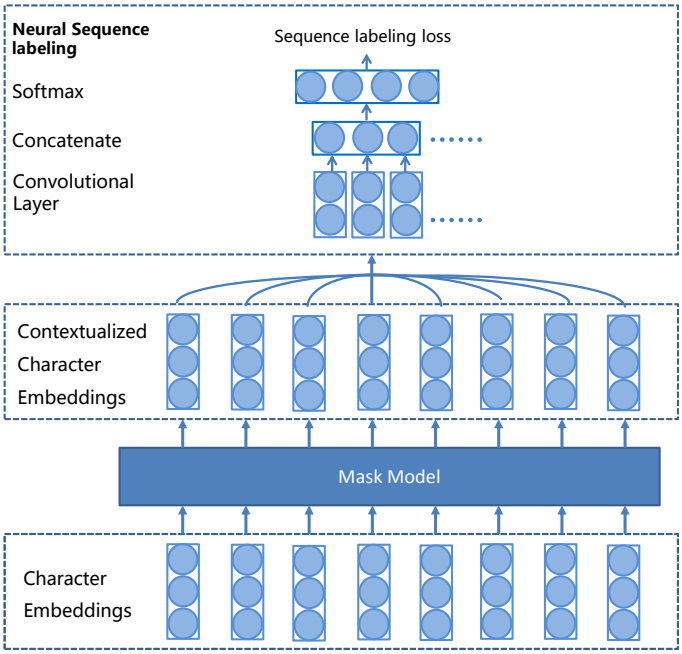


**Fig. 1.** The proposed MaskCNN.

only reserve the convolutional layer. And the convolutional neural segmenter is equivalent to a feed-forward neural network [9] with multiple window sizes. We decide to take convolutional neural segmenter as our baseline model, due to that (1) convolutional neural segmenters follow most of the word segmentation systems working by sequence labelling [32]; and (2) previous baseline segmenters [20, 35, 19] are limited with local features. Therefore, it may be unfair to take recurrent networks with context information as rival; (3) the performance of convolutional neural segmenter is comparable with previous baseline segmenters.

The basic unit of convolutional neural networks (CNN) is filters [17]. Take contextualized embedding $c$ as the input. Then features $c_i^{'}$ from a filter $i$ is generated by:

$$c_i^{'} = f(m \otimes c + b), \tag{8}$$

where $\otimes$ is convolution operator, $m$ and $b$ are the weight matrix and bias, $f$ is ReLU in our network. And for each window size, we apply multiple filters to generate multiple feature maps. Features from different feature maps are concatenated. Then a softmax layer is appended for predicting the label of each character. Our neural word segmenter regards Chinese word segmentation as a sequence labelling task. The segmenter adopts BIES (Begin, Inside, End, Single) four labels scheme which represents the position of character inside a word. During the training phase, the cross-entropy cost function is used. And during the testing phase, the label sequences are constructed through beam search.

### 2.4 Train Strategy

In this section, we propose our neural domain adaptation method based on the contextualized character embedding and name it as MaskCNN. As shown in Fig. 1, the character sequence auto-encoder and the neural segmenter share the same contextualized character embedding layer.

In our method, first, unlabelled data from both *source* and *target* domains are employed to train the contextualized character embedding layer and character sequence auto-encoder through auto-encoder rebuilding loss.[1] During this step, the characteristics of both domains are stored in the contextualized character embedding layer and the contextualized character embedding layer is learned to resolve the character ambiguous by considering surrounding characters.

Then the contextualized character embedding is kept fixed to avoid domain-specific training, and the neural segmenter is trained by *source* domain annotated data. In this step, the segmenter is trained to do word segmentation based on the contextualized character embedding.

## 3 Experiments

### 3.1 Dataset

In this paper, the Chinese Treebank (CTB) [31] is selected as the *source* domain data. We train our model by using the annotated data from CTB. The Patent [18] and Zhuxian [35] are used as the *target* domain data. A patent is often a description of a system or solution to a specific technology problem. Patents often contain a high concentration of technical terms which are rare in daily newswire. Zhuxian is a Internet novel which is written in a different style from newswire and contains many novel specific named entity. Some unlabeled target data is used to generate the contextualized character embeddings with the source data. We use the trained model to segment patent and novel data. The statistics of the data is shown in Table 1. We compare our proposed model with methods mentioned in [19, 35] which are feature-based and lexicon-based methods. We use same amount of unlabelled *target* domain data as previous methods [19, 35] to make a fair comparison.

### 3.2 Hyper-Parameter Settings

In the experiments, the hyper parameters are chosen according to the balance of development data performance and training time. For the segmenter, the window size of filters is set as 2,3,4,5 and feature maps of each window size are 300 with 50% dropout. For mask network, the window size is 5 and the hidden units have the same size as embeddings with no dropout. The size of character and bigram embeddings is 200 with 20% dropout. The bigram embedding is used following the implements of [36]. The hidden unit of LSTM sequence to sequence auto-encoder is 1000 with 50% dropout. The training is done through stochastic gradient descent with a batch size of 16 and Adadelta

[1] We try to weight the *target* domains data more, but no significant improvement is observed.

| Type | Sec. | Source | | Sec. | Target | |
|---|---|---|---|---|---|---|
| | | CTB5 | CTB7 | | Patent | Zhuxian |
| sent. | train | 18k | 36k | unlabel | 11k | 16k |
| words. | | 641k | 839k | | - | - |
| sent. | dev. | 0.35k | 4.8k | dev. | 1.5k | 0.79k |
| words. | | 6.8k | 120k | | 46.2k | 20.4k |
| sent. | test | 0.35k | 11k | test | 1.5k | 1.4k |
| words. | | 8.0k | 241k | | 48.4k | 34.4k |

**Table 1.** Statistics of source and target datasets

| | CTB5 → Zhuxian | | CTB7 → Patent | |
|---|---|---|---|---|
| unigrams | | | | |
| num | 4.3k | 2.1k | 4.5k | 1.6k |
| diff | 106 (4.9%) | | 69 (4.4%) | |
| cover | 99.36% | | 99.45% | |
| bigrams | | | | |
| num | 175k | 25k | 240k | 29k |
| diff | 13k (51.7%) | | 14k (47.9%) | |
| cover | 67.61% | | 68.88% | |

**Table 2.** Statistics of different domains. *num* is the amount of unique uni/bigrams. *diff* is the number of *target* domain specific unique uni/bigrams. *cover* is the percentage of uni/bigrams in *target* domain that can be covered by *source* domain.

update rule [33]. The beam size of beam search is 10. We per-train the embeddings using the publicly available Chinese Wikipedia corpus which is 822 MB and contains about 11 million sentences[2]. The embedding vectors are pre-trained using *word2vec* with the continuous skip-gram architecture.

### 3.3 Differences between Different Domain

In order to explore the differences between different domains of Chinese word segmentation, we count up the unigrams and bigrams in different domains and the statistics of different domains are listed in Table 2. From the table, we find that almost all of the unigrams in *target* domain are already available in *source* domain, but more than 30% of the bigrams only appear in *target* domain. It is obvious that the main difference between different domains is bigrams or the contextual characters. This inspires us to model the contextualization of characters in different domains through a mask model.

### 3.4 Main Results

[2] `http://download.wikipedia.com/zhwiki/latest/`

| Methods | P | R | F1 |
|---|---|---|---|
| (Zhang et al., 2014)[35] | | | |
| Baseline | - | - | 87.71 |
| +Self-Training | - | - | 88.62 |
| Ours | | | |
| Baseline | 85.91 | 85.05 | 85.48 |
| MaskCNN | 87.76 | 87.83 | 87.80 |
| MaskCNN+bigram | 89.75 | 88.95 | 89.35 |

**Table 3.** The results between CTB5 and Zhuxian

| Methods | P | R | F1 |
|---|---|---|---|
| (Li and Xue, 2016)[19] | | | |
| Baseline | 86.10 | 86.30 | 86.20 |
| Baseline+Features | 89.17 | 88.59 | 88.88 |
| Ours | | | |
| Baseline | 86.31 | 86.30 | 86.31 |
| MaskCNN | 87.98 | 86.89 | 87.43 |
| MaskCNN+bigram | 88.40 | 87.87 | 88.14 |

**Table 4.** The results between CTB7 and Patent

**From CTB5 to Zhuxian** We first compare our methods with the methods from [35] for the adaptation from CTB5 to Zhuxian. The baseline model of [35] is a discriminative joint segmentation and tagging model. Our baseline model is the convolutional neural segmenter. A self-training method is adopted to extend training data by automatically labelling target sentences with the *source* domain training data. The other methods mentioned in [35] used a domain-specific lexicon. As our proposed method do not use the lexicon in this paper, we do not list the results of the other methods mentioned in [35] for comparison.

The results are shown in Table 3. The MaskCNN model achieves an improvement over the baseline by 2.32 absolute percentage. With the bigram embeddings, the result of our proposed method is better than the result of the self-training method.

**From CTB7 to Patent** We also examine the performance of our method for the adaptation from CTB7 to Patent. We compare our method with the method from [19]. Li and Xue [19] proposed in-domain features and out-of-domain features, which are manual-crafted features, to improve the performance of Chinese patent word segmentation. As mentioned in their paper, the out-of-domain features are extracted to share common characteristics across *source* and *target* domains. The *Out-of-domain features* includes *character POS feature* (C_POS), *word dictionary feature* (Dict) and *character similarity feature* (Sim).

The results are shown in Table 4. In paper [19], the baseline model uses a CRFs model with basic features. From the results, we can see that the performance of our

baseline model is comparable to the performance of their baseline model. Our proposed MaskCNN model achieves an improvement over the baseline by 1.12 absolute percentage. Then the MsakCNN model obtains the advanced improvement with the help of bigram embeddings. Although our proposed model does not outperform the best model from the paper [19], the result is very close. As our neural domain adaptation method does not depend on the lexicon, it is much easier for applying without any restriction.

## 4 Related Work

Domain adaptation can be roughly divided into two scenarios, the fully supervised domain adaptation and the semi-supervised domain adaptation [10]. The *easy domain adaptation* is well known for fully supervised scenario. The feature space is first augmented of both *source* and *target* data and then the combined feature space is used to train cross-domain model [10]. But obtaining annotated data could be expensive, and it would be a huge cost to annotate data for every domain. Many semi-supervised domain adaptation methods are proposed in tasks such as the sentiment classification. The main idea is the unsupervised learning of a general representation that works in both domains. Both feature-based [1, 22] and neural-based [13, 2] semi-supervised domain adaptation methods were explored. There are other divisions for domain adaptation, for example, this problem can be divided into token-supervised and type-supervised methods [35].

Recently, neural network models had been increasingly investigated in Chinese word segmentation for their ability of automatic feature representation [9, 38, 23, 6, 5, 30, 36, 3]. These neural models alleviated the burden of manual feature engineering and achieved competitive performance with the hand-crafted models.

In the domain adaptation of Chinese word segmentation, previous works mainly focused on feature-based and lexicon-based methods. Unsupervised character clustering and self-training method were applied [20]. Manually annotated lexicons and sentences achieved significant improvement [35] while partially-annotated data was proved to be more effective [21]. Li and Xue [18, 19] annotated a significant amount of Chinese patent data and designed features to capture the distributional characteristics in patents. Qiu and Zhang [25] mined entities in Chinese novel with information extraction techniques.

Choi et al. [8] use the context-dependent word representation to improve the performance of machine translation. In their work, the contextualization disambiguates the meaning of the word by masking out some dimensions of the word embedding vectors based on the context.

## 5 Conclusion

In this paper, we focus on the semi-supervised scenario of domain adaptation and explore method to adapt the information cross different domains with different document types for neural Chinese word segmentation. We first introduce a mask model to obtain the contextualized character embedding. Then the neural segmenter works above the contextualized character embedding.

In the experiments, we explore the differences between different domains. Experiments show that although previous feature-based and lexicon-based methods are strong domain adaptation methods, our neural domain adaptation method achieves competitive performance without additional lexicons.

## Acknowledge

This work was supported by Beijing Natural Science Foundation (4174098), National Natural Science Foundation of China (61702047) and the Fundamental Research Funds for the Central Universities (2017RC02).

## References

1. Blitzer, J., McDonald, R., Pereira, F.: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, chap. Domain Adaptation with Structural Correspondence Learning, pp. 120–128. Association for Computational Linguistics (2006), http://aclweb.org/anthology/W06-1615
2. Bollegala, D., Maehara, T., Kawarabayashi, K.i.: Unsupervised cross-domain word representation learning. 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). pp. 730–740. Association for Computational Linguistics (2015), http://aclweb.org/anthology/P15-1071
3. 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 (2016), http://aclweb.org/anthology/P16-1039
4. Chen, X., Qiu, X., Huang, X.: A long dependency aware deep architecture for joint chinese word segmentation and pos tagging. arXiv preprint arXiv:1611.05384 (2016)
5. Chen, X., Qiu, X., Zhu, C., Huang, X.: Gated recursive neural network for 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 1: Long Papers). pp. 1744–1753. Association for Computational Linguistics (2015), http://aclweb.org/anthology/P15-1168
6. Chen, X., Qiu, X., Zhu, C., Liu, P., Huang, X.: Long short-term memory neural networks for chinese word segmentation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1197–1206. Association for Computational Linguistics (2015), http://aclweb.org/anthology/D15-1141
7. Chen, X., Shi, Z., Qiu, X., Huang, X.: Adversarial multi-criteria learning for chinese word segmentation. arXiv preprint arXiv:1704.07556 (2017)
8. Choi, H., Cho, K., Bengio, Y.: Context-dependent word representation for neural machine translation. Computer Speech and Language (2016)
9. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. The Journal of Machine Learning Research 12, 2493–2537 (2011)
10. Daume III, H.: Frustratingly easy domain adaptation. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics. pp. 256–263. Association for Computational Linguistics (2007), http://aclweb.org/anthology/P07-1033

11. Daume III, H., Marcu, D.: Domain adaptation for statistical classifiers. Journal of Artificial Intelligence Research 26, 101–126 (2006)
12. Emerson, T.: The second international chinese word segmentation bakeoff. In: Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing (2005), http://aclweb.org/anthology/I05-3017
13. Glorot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: A deep learning approach. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11). pp. 513–520 (2011)
14. Hatori, J., Matsuzaki, T., Miyao, Y., Tsujii, J.: Incremental joint approach to word segmentation, pos tagging, and dependency parsing in chinese. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1045–1053. Association for Computational Linguistics (2012), http://aclweb.org/anthology/P12-1110
15. Jiang, W., Huang, L., Liu, Q.: Automatic adaptation of annotation standards: Chinese word segmentation and pos tagging – a case study. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. pp. 522–530. Association for Computational Linguistics (2009), http://aclweb.org/anthology/P09-1059
16. Kiat Low, J., Tou Ng, H., Guo, W.: A maximum entropy approach to chinese word segmentation. In: Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing (2005), http://aclweb.org/anthology/I05-3025
17. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1746–1751. Association for Computational Linguistics (2014)
18. Li, S., Xue, N.: Effective document-level features for chinese patent word segmentation. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 199–205. Association for Computational Linguistics (2014), http://aclweb.org/anthology/P14-2033
19. Li, S., Xue, N.: Towards accurate word segmentation for chinese patents. arXiv preprint arXiv:1611.10038 (2016)
20. Liu, Y., Zhang, Y.: Unsupervised domain adaptation for joint segmentation and pos-tagging. In: Proceedings of COLING 2012: Posters. pp. 745–754. The COLING 2012 Organizing Committee (2012), http://aclweb.org/anthology/C12-2073
21. Liu, Y., Zhang, Y., Che, W., Liu, T., Wu, F.: Domain adaptation for crf-based chinese word segmentation using free annotations. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 864–874. Association for Computational Linguistics (2014), http://aclweb.org/anthology/D14-1093
22. Pan, S.J., Ni, X., Sun, J.T., Yang, Q., Chen, Z.: Cross-domain sentiment classification via spectral feature alignment. In: International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, Usa, April. pp. 751–760 (2010)
23. Pei, W., Ge, T., Chang, B.: 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). pp. 293–303. Association for Computational Linguistics (2014), http://aclweb.org/anthology/P14-1028
24. Peng, F., Feng, F., McCallum, A.: Chinese segmentation and new word detection using conditional random fields. In: COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics (2004), http://aclweb.org/anthology/C04-1081
25. Qiu, L., Zhang, Y.: Word segmentation for chinese novels. In: AAAI. pp. 2440–2446 (2015)
26. Sun, W.: A stacked sub-word model for joint chinese word segmentation and part-of-speech tagging. In: Proceedings of the 49th Annual Meeting of the Association for Computational

Linguistics: Human Language Technologies. pp. 1385–1394. Association for Computational Linguistics (2011), http://aclweb.org/anthology/P11-1139
27. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: International Conference on Neural Information Processing Systems. pp. 3104–3112 (2014)
28. 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 (2005), http://aclweb.org/anthology/I05-3027
29. Weaver, W.: Translation. John Wiley and Sons (1949)
30. Xu, J., Sun, X.: Dependency-based gated recursive neural network for chinese word segmentation. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 567–572. Association for Computational Linguistics (2016), http://aclweb.org/anthology/P16-2092
31. Xue, N., Xia, F., Chiou, F.D., Palmer, M.: The penn chinese treebank: Phrase structure annotation of a large corpus. Natural language engineering 11(02), 207–238 (2005)
32. Xue, N.: Chinese word segmentation as character tagging. In: International Journal of Computational Linguistics & Chinese Language Processing, Volume 8, Number 1, February 2003: Special Issue on Word Formation and Chinese Language Processing. pp. 29–48 (2003), http://aclweb.org/anthology/O03-4002
33. Zeiler, M.D.: Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)
34. Zeng, X., Wong, F.D., Chao, S.L., Trancoso, I.: Co-regularizing character-based and word-based models for semi-supervised chinese word segmentation. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 171–176. Association for Computational Linguistics (2013), http://aclweb.org/anthology/P13-2031
35. Zhang, M., Zhang, Y., Che, W., Liu, T.: Type-supervised domain adaptation for joint segmentation and pos-tagging. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics. pp. 588–597. Association for Computational Linguistics (2014), http://aclweb.org/anthology/E14-1062
36. Zhang, M., Zhang, Y., Fu, G.: Transition-based neural word segmentation. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 421–431. Association for Computational Linguistics (2016), http://aclweb.org/anthology/P16-1040
37. Zhang, Y., Clark, S.: Joint word segmentation and pos tagging using a single perceptron. In: Proceedings of ACL-08: HLT. pp. 888–896. Association for Computational Linguistics (2008), http://aclweb.org/anthology/P08-1101
38. Zheng, X., Chen, H., Xu, T.: Deep learning for chinese word segmentation and pos tagging. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 647–657. Association for Computational Linguistics (2013), http://aclweb.org/anthology/D13-1061