ChinaRxiv

AI translation · View original & related papers at
chinarxiv.org/items/chinaxiv-201812.00101

Postprint: Research on Multi-Model Chinese Word Segmentation Methods Based on Character Clusters

Authors: Li Duihong, Wang Peiyan, Zhang Guiping, Zhang Shaoyang

Date: 2018-12-13T00:00:00+00:00

Abstract

Character-based tagging segmentation methods represent a relatively effective approach in the current field of Chinese word segmentation. However, since Chinese characters inherently carry semantic information, and different characters possess distinct meanings and functions across various contexts, the word formation patterns of each character exhibit significant variation. To address this issue, we propose a multi-model Chinese word segmentation method based on character clusters. This method first models each individual character, then performs cluster analysis on the learned model parameters to form character clusters, and finally retrains the model parameters based on these clusters. Experimental results demonstrate that the proposed method can effectively identify character clusters with identical or similar word formation patterns, and successfully distinguishes the varying degrees of influence that similar features exert on different characters.

Full Text

Preamble

Multi-model Chinese Word Segmentation Method Based on Character Clusters

Li Duihong, Wang Peiyan†, Zhang Guiping, Zhang Shaoyang
(Human-Computer Intelligence Research Center, Shenyang Aerospace University, Shenyang 110136, China)

Abstract: Character-based tagging methods are currently effective approaches in Chinese word segmentation. However, Chinese characters inherently carry semantic information, and different characters serve different meanings and functions in various contexts, leading to differences in word-formation patterns

for each character. To address this issue, this paper proposes a multi-model Chinese word segmentation method based on character clusters. The method first models each character individually, then performs cluster analysis on the learned model parameters to form character clusters, and finally retrains model parameters based on these clusters. Experimental results demonstrate that this method can effectively discover character clusters with identical or similar word-formation patterns and successfully distinguishes the varying degrees of influence that similar features exert on different characters.

Keywords: Chinese word segmentation; word-formation rules; model parameters; clustering

0 Introduction

Words are the smallest linguistic units that can be used independently. Unlike English and other Western languages, Chinese uses characters as the basic writing unit without explicit delimiters between words. Without segmentation, computers cannot determine the exact boundaries of Chinese words and consequently struggle to comprehend the semantic information contained in texts [1]. Therefore, Chinese word segmentation is a fundamental task in natural language processing, playing a pivotal role in named entity recognition, automatic text classification, machine translation, and other areas. Its performance directly impacts subsequent natural language processing tasks.

Among Chinese word segmentation methods, supervised character-based tagging approaches [2] have demonstrated favorable segmentation effectiveness. These methods abstract the segmentation process as a sequence labeling task and employ machine learning models suitable for sequence labeling. Widely used sequence labeling models include Maximum Entropy Markov Models (MEMM) [3], Hidden Markov Models (HMM) [4], and Conditional Random Fields (CRF) [5-7]. However, the performance of these models is largely constrained by feature selection and extraction. In recent years, with the vigorous development of deep learning, neural network models such as Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) networks, and their variants—well-suited for sequence labeling tasks—have been extensively applied to word segmentation [8-11].

Both traditional machine learning models and neural network models focus on each character (or token) in a sentence, determining its positional information in word formation based on the context of the character to be labeled, which serves as the segmentation tag. These methods establish a single set of model parameters based on training corpora, considering the global combined effect of context on all characters. In other words, they assume that identical contextual environments exert the same influence on different characters to be labeled, learning general patterns of character word formation. However, because Chinese characters inherently carry semantic information, each character exhibits

distinct word-formation patterns. Even when the same character serves as a contextual feature for the character to be labeled, its meaning and function can vary significantly [12,13], altering the degree of cohesion with the character to be labeled. Consider the following examples:

a) 建立/稳定/和睦/的/两岸/关系/。/

b) 营造/了/民主/和谐/的/气氛/。/

c) 党中央/坚持/领导/和/党/的/十五大/精神/。/

In these three examples, when the current characters to be labeled are "睦" (mù), "谐" (xié), and "党" (dǎng) respectively, the preceding feature is "和" (hé) in all cases. Yet the same feature has different effects on the characters to be labeled, meaning the degree of cohesion varies. As the examples show, the "和" in the first two sentences exhibits the same degree of cohesion with "睦" and "谐," while the "和" in the third sentence shows a different degree of cohesion with "党" compared to the first two examples. Therefore, the assumption that contextual environments have identical effects on characters to be labeled is clearly problematic.

To address this issue, reference [14] proposed a character-based multi-model Chinese word segmentation method. This method' s key characteristic is constructing separate model parameters for each character, effectively distinguishing the influence of identical features on different characters to be labeled and learning specific patterns of character word formation. However, this approach has limitations. Although model parameters trained for each character can effectively reflect that character' s word-formation patterns, certain characters share identical or similar word-formation patterns, inevitably leading to redundant model parameters. Additionally, some characters have limited training samples, which can reduce out-of-vocabulary (OOV) word recall.

This paper improves upon the character-based multi-model Chinese word segmentation method [14] by proposing a multi-model segmentation approach based on character clusters. This method performs cluster analysis on the model parameters learned from the character-based multi-model approach, aggregating characters with identical or similar word-formation patterns into clusters, and then retrains model parameters based on these clusters. Compared to single-model methods, this approach effectively improves in-vocabulary (IV) word recall, and compared to multi-model methods, it enhances OOV word recall. These improvements have been validated on both PKU and MSR corpora.

1.1 Model Training Process

This paper proposes a multi-model word segmentation method based on character clusters. The method first trains model parameters for each character using

the character-based multi-model approach. These character model parameters represent the word-formation patterns of each character. Next, cluster analysis is performed on these model parameters to discover the intrinsic distribution structure among them, aggregating characters with identical or similar word-formation patterns into clusters. Finally, model parameters are retrained based on these clusters. The specific training process consists of three components: character model parameter acquisition, discovery of character word-formation pattern distribution structure, and model retraining, as shown in [Figure 1: see original paper].

1.2 Model Structure

Chinese word segmentation is typically treated as a character-level sequence labeling problem. Therefore, the segmentation process can be viewed as a machine learning process that labels each character in a string. Drawing inspiration from Ma Jianqiang et al. [15], this paper divides the model structure into three components: a Look-up table, a Concatenation function, and a Sigmoid function. Unlike the character-based multi-model approach, our segmentation method models based on character clusters. During segmentation, decisions are made according to the corresponding cluster model parameters. To reduce problem complexity, each cluster model adopts an identical structure. The specific model structure is illustrated in [Figure 2: see original paper].

a) Look-up Table: This records the mapping relationship between features and real-valued vectors, also known as feature embeddings. The embedding for each distinct feature t is denoted as ⟦MATH_{EMBED}T⟧, where N represents the dimensionality of the real-valued vector. Features are extracted from the training corpus.

b) Concatenation Function: To predict the tag status of the character to be labeled, the embeddings of its corresponding features must be concatenated into a single vector as model input, denoted as ⟦MATH_{CONCAT}⟧, where K is the number of features used to describe the character to be labeled, and N is the dimensionality of feature embeddings.

c) Sigmoid Function: The activation function employed in the model structure is the Sigmoid function, defined as in equation (1). Here, a represents the input feature embedding, w denotes feature weights, and ⟦MATH_{DOT}⟧ indicates the dot product of two vectors.

1.2.1 Input

This paper extracts features from a context window of width 5, including both unigram and bigram features, as shown in . Previous research has demonstrated that a 5-character context window roughly captures the context of one word before and after [16], meaning it encompasses both character-level and word-level

information, sufficient to cover the vast majority of word-formation scenarios in real text.

** Uni- and Bi-gram Feature Template**

The subscripts represent the relative position of characters to the character to be labeled. *C_i* denotes the current character to be labeled, *C_{i-1}* represents the preceding character, *C_{i+1}* represents the following character, and so forth. Using “睦” and “谐” from the previous examples as current characters to be labeled, the unigram features for “睦” are “定”, “和”, “睦”, “的”, “两”, and the corresponding bigram features are “定和”, “和睦”, “睦的”, “的两”. Similarly, for “谐”, the unigram features are “主”, “和”, “谐”, “的”, “气”, and the bigram features are “主和”, “和谐”, “谐的”, “的气”.

1.2.2 Output

Each character in a character sequence receives a specific word-position label. This paper uses two tags, “S” and “C”, to represent possible tag statuses for the current character to be labeled. The “S” (Separation) tag indicates that the current character is in a separated state from the previous character, meaning a new word begins with the current character. The “C” (Combination) tag indicates that the current character is in a combined state with the previous character, forming a word or part of a word. The following sentence demonstrates the correct tag sequence:

建-S 立-C 稳-S 定-C 和-S 睦-C 的-S 两-S 岸-C 关-S 系-C 。-S

The activation function output values in this paper are distributed within (0,1). A threshold of 0.5 is used: output values greater than 0.5 are labeled as “S”, otherwise as “C”.

1.3 Model Training

This paper employs cross-entropy as the loss function, as shown in equation (2). The training process first predicts the tag of the character to be labeled under current parameters, then updates model parameters based on its correct tag in the corpus.

To prevent overfitting and degradation of model generalization ability, this paper adds an L2 regularization term to the loss function, as shown in equation (3). Here, λ is the regularization coefficient controlling the strength of regularization.

This paper uses stochastic gradient descent to optimize the objective function, employing backpropagation to compute gradients of the objective function with respect to a and w, updating them while keeping the other unchanged. The update formulas are shown in equations (4) and (5).

1.4 Clustering-Based Discovery of Word-Formation Pattern Distribution Structure

In unsupervised learning, clustering algorithms can be used to discover intrinsic distribution structures in data. This paper employs hierarchical clustering [17] as a precursor to subsequent model training to discover the word-formation pattern distribution structure represented by model parameters in the character-based multi-model approach. Hierarchical clustering initially treats each model parameter as a separate cluster, then continuously merges these atomic clusters based on distance to form a tree-structured clustering hierarchy. Finally, the hierarchy is partitioned according to a predefined inter-cluster splitting criterion to form final clusters. The specific algorithmic process is as follows:

Algorithm Steps: a) Treat each model parameter as a class and compute pairwise distances;
b) Merge the two classes with the smallest distance into a new class;
c) Recalculate distances between the new class and all other classes;
d) Repeat steps (2) and (3) to generate a tree-structured clustering hierarchy;
e) Partition the hierarchy according to the inter-cluster splitting criterion to form final clusters.

Hierarchical clustering typically employs Euclidean distance or cosine similarity as distance metrics, as shown in equations (6) and (7). Here, x and x' represent model parameters, d denotes vector dimensionality. Euclidean distance measures absolute positional distance between vectors, with smaller values indicating closer proximity and greater similarity between model parameters. Cosine similarity measures the angle between vectors, with larger values indicating smaller angles and closer proximity between model parameters.

During algorithm execution, to uniformly adopt the minimum distance as the cluster merging condition, when using cosine similarity as the distance metric, the actual implementation employs *1 - cosine_{similarity}*, where smaller values indicate greater similarity between model parameters.

The inter-cluster splitting criterion uses the inconsistency coefficient, which reflects the degree of inconsistency between the distance at which two clusters merge in the tree structure and the distances at which clusters at depth 2 below them merge. When partitioning into clearly distinct clusters, the inconsistency coefficient is high, and vice versa. The inconsistency coefficient is calculated as in equation (8), where h represents the inconsistency coefficient, d denotes the distance between two merging clusters, avg represents the mean distance of cluster merges at depth 2 below, and std denotes the standard deviation of these distances. When the inconsistency coefficient of two clusters is below the threshold, they are merged into a new cluster; otherwise, they are split.

1.5 Performance Analysis of Segmentation Algorithm

The performance of segmentation algorithms is typically measured by computational complexity, including time complexity and space complexity. In machine learning, since the training process is a one-time operation and segmentation does not require retraining, less attention is paid to time costs during training. In practical segmentation, greater emphasis is placed on segmentation speed and model storage space.

This section focuses on analyzing the time complexity of the segmentation process. Segmentation requires looking up the Look-up table and model parameters w generated during training. Given m characters to be segmented and n entries in the Look-up table and model parameters, the lookup time complexity is O(1). Since each character' s labeling requires traversing the entire Look-up table and model parameters w, the time complexity of the segmentation process is O(mn). Compared to the character-based multi-model approach, the proposed method has far fewer models, thus offering advantages in both time and space complexity.

2.1 Data and Preprocessing

The corpora used in our experiments are the PKU and MSR corpora provided by the second International Chinese Word Segmentation Bakeoff 2005 organized by SIGHAN. These include training sets, test sets, gold standard answers for test sets, dictionaries, and scoring scripts. Detailed corpus information is shown in .

** Corpus Details of PKU and MSR**

When extracting features from a context window of width 5 around the current character, padding is required for sentence-initial or sentence-final characters. Two special characters such as “start-1” , “start-2” (or “end-1” , “end-2”) are added to the left (or right) of the character, with their tag status set to “S” .

2.2 Evaluation Methods

Chinese word segmentation performance is typically evaluated using precision (P), recall (R), F-score (F), out-of-vocabulary recall (ROOV), and in-vocabulary recall (RIV). The F-score serves as the primary reference metric for segmentation performance, while OOV recall effectively reflects model generalization ability.

2.3 Experimental Parameter Settings

Hyperparameter selection in neural network models significantly impacts segmentation performance. The hyperparameter settings are shown in . Previous work [1] has demonstrated through extensive experiments that setting feature embedding dimensionality to 50 can ensure both training speed and segmentation performance. Therefore, this paper sets feature embedding dimensionality to 50. Additionally, to prevent training failure due to potential errors, two termination conditions are set: maximum iteration count and loss tolerance. Training terminates when either condition is met.

** Setting of the Hyper-parameters**

2.4 Experimental Results and Analysis

To validate the effectiveness of the proposed method, we compare it with several segmentation approaches: CRF-based methods, single-model methods, character-based multi-model methods, and neural network methods. The single-model approach builds a single set of model parameters for the training corpus. The character-based multi-model approach constructs separate model parameters for each character. CRF methods employ three strategies: 2-tag and 4-tag schemes with tag bigram transition features (denoted CRF2 and CRF4), and a 2-tag scheme without tag bigram transition features (denoted CRF). All CRF experiments use the feature template shown in , with feature embeddings randomly initialized for neural network model comparisons.

2.4.1 Experiments on Distance Metrics and Threshold Selection for Clustering

Hierarchical clustering is applied to character models trained via the character-based multi-model approach, aiming to aggregate characters with identical or similar word-formation patterns into clusters. The quality of clustering results directly impacts subsequent model retraining. Tables 4 and 5 show experimental results using different distance metrics.

** Performances of Using Different Distance Metrics in PKU Test Set**

** Performances of Using Different Distance Metrics in MSR Test Set**

The results demonstrate that cosine similarity yields the best segmentation performance on both corpora. Cosine similarity measures the angle between vectors, reflecting their similarity, whereas Euclidean distance measures absolute positional distance. Since our clustering objects are model parameters that reflect how contextual features influence character labeling status—representing word-formation patterns—cosine similarity is more appropriate. shows clustering results using cosine similarity.

** Characters Similar to “吴”, “鸯”, “扒”, “蚣” **

As shown in , “吴” is a surname, and the similar characters obtained through clustering—“赵”, “彭”, “徐”, “卢”, “蔡”—are also surnames with identical word-formation patterns. Similarly, “鸯”, “扒”, and “蚣” correspond to bird names, verbs, and insect names respectively, each forming clusters with characters from corresponding categories. This demonstrates that clustering model parameters from the character-based multi-model approach effectively obtains character clusters with identical or similar word-formation patterns.

Unlike K-Means, hierarchical clustering does not require pre-specifying the number of clusters but instead uses an inconsistency coefficient threshold to obtain optimal clusters. Therefore, threshold setting affects experimental results. [Figure 3: see original paper] shows performance using different inconsistency coefficients.

[Figure 3: see original paper] Performance of Using Different Inconsistency Coefficients

The results indicate that on both corpora, hierarchical clustering of character-based multi-model parameters achieves optimal clustering and segmentation performance when the inconsistency coefficient threshold is set to 1. This setting is used in subsequent experiments.

2.4.2 Model Comparison Experiments

Tables 7 and 8 compare the proposed method with single-model and character-based multi-model approaches. The proposed method demonstrates superior segmentation performance on both corpora. On the PKU corpus, the F-score is 1.2 percentage points higher than the single-model method and 0.1 percentage points higher than the multi-model method. On the MSR corpus, the F-score exceeds the single-model method by 4.4 percentage points and the multi-model method by 0.1 percentage points, demonstrating sufficient stability. The single-model method shows clear advantages in OOV recognition by considering global contextual effects and learning general word-formation patterns, while the multi-model method excels in IV recall by modeling each character individually and learning specific patterns. Our method combines both approaches through parameter clustering, learning both general and specific word-formation patterns, thus outperforming both baselines.

** Comparison with Performance on PKU Corpus**

** Comparison with Performance on MSR Corpus**

We also compare model quantities among the three methods, as shown in . Compared to the multi-model approach, model count decreases from 4,686 to 1,854 on PKU corpus (nearly a three-fifths reduction) and from 5,151 to 2,299 on MSR corpus (nearly a one-half reduction). This demonstrates that our method significantly reduces model storage costs while improving F-score.

** Comparison with Model Numbers on Two Corpora**

To further validate usefulness, we compare segmentation time and model storage space among the three methods. Segmentation time refers to the time consumed during segmentation using trained models; storage space refers to the size of model storage. Results are shown in .

** Comparison with Word Segmentation Time and Model Storage Space on Two Corpora**

The results show that compared to the multi-model method, our approach offers advantages in both segmentation speed and storage space, particularly in storage space, significantly reducing costs and facilitating practical engineering applications. The single-model method uses minimal storage and has the fastest segmentation speed by building a single parameter set. The character-based multi-model method suffers from redundancy by modeling each character separately. Our method merges characters with identical or similar word-formation patterns into clusters, substantially reducing model count while improving performance and decreasing storage requirements.

Tables 11 and 12 compare our method with CRF-based approaches. CRF4 with 4 tags and transition features shows strong performance. On PKU corpus, our method outperforms CRF4 by 0.4 percentage points in F-score across all five metrics. However, on MSR corpus, CRF4 surpasses our method by 0.8 percentage points, primarily due to significantly higher OOV recall, likely because the larger MSR corpus enables more thorough training.

** Comparison with Results Using CRF on PKU Corpus**

** Comparison with Results Using CRF on MSR Corpus**

We also compare our results with previous work on the same datasets, including Zheng et al. (2013) using Collobert et al.' s neural network framework, Pei et al. (2014) with MMTNN using tag embeddings and tensor-based transitions, Chen et al. (2015) with LSTM for long-term dependencies, and Cai et al. (2016) using gated compositional neural networks for character representations with LSTM scoring. As shown in , our method achieves F-score improvements of 1.1 and 2.3 percentage points over Zheng et al. on PKU and MSR corpora respectively, and matches or exceeds Pei et al.' s results.

** Comparison with Previous Models**

Further analysis reveals our method performs better on segmenting single-character verbs like "入/军队" (enter/army), "服/现役" (serve/active duty), "战/风雪" (battle/wind and snow), "拟/任" (plan/appoint), and "求/发展" (seek/development) compared to Cai et al. However, error analysis also identifies segmentation errors such as those listed in .

** Comparison with Segmentation Results**

The errors primarily involve multi-word 粘连 (adhesion), a issue discussed in

literature [23], which demonstrates that character-based methods often ignore compositional information within words and that character-word joint decoding yields better results. Our method lacks word-level information guidance, leading to such adhesion errors. In contrast, Cai et al. introduce word information through gated compositional neural networks for candidate word distributed representations and score all segmentation combinations with LSTM, selecting the highest-scoring combination. Future work will incorporate word information following Cai et al.' s approach.

3 Conclusion

This paper proposes a multi-model Chinese word segmentation method based on character clusters, combining the strengths of single-model and character-based multi-model approaches. It leverages the single-model' s ability to discover OOV words and the multi-model' s strength in segmenting IV words, learning both general and specific word-formation patterns. Experimental results show that compared to the character-based multi-model method, our approach slightly improves segmentation performance while significantly reducing model count, storage costs, and segmentation time.

However, our method does not incorporate word-level information to guide the segmentation process, limiting final performance. Future work will explore integrating word information and investigate alternative algorithms such as multi-task learning [24-26] for discovering word-formation pattern distribution structures, as clustering quality directly impacts segmentation effectiveness.

References

[1] Lai Siwei, Xu Liheng, Chen Yubo, et al. Exploring Chinese word segmentation algorithm based on representation learning [J]. Journal of Chinese Information Processing, 2013, 27(5): 8-14.

[2] Xue Neiwen, Shen Libin. Chinese word segmentation as LMR tagging [C]//Proc of the 2nd SIGHAN Workshop on Chinese Language Processing. New York: ACM Press, 2003: 176-179.

[3] McCallum A, Freitag D, Pereira F. Maximum entropy markov models for information extraction and segmentation [C]//Proc of International Conference on Machine Learning. New York: ACM Press, 2000: 591-598.

[4] Li Yuelun, Chang Baobao. Chinese word segmentation method based on maximum interval markov network model [J]. Journal of Chinese Information Processing, 2010, 24(1): 8-14.

[5] Tseng H, Chang P, Andrew G, et al. A conditional random field word segmenter for sighan bakeoff 2005 [C]//Proc of the 4th SIGHAN Workshop on Chinese Language Processing. New York: ACM Press, 2005: 168-171.

[6] Zhang Ruiqiang, Kikui G, Sumita E. Subword-based tagging by conditional random fields for Chinese word segmentation [C]//Proc of Human Language Technology Conference of the North American Chapter of the ACL. Stroudsbury, PA: ACL, 2006: 193-196.

[7] Zhao Hai, Huang Changning, et al. Effective tag set selection in Chinese word segmentation via conditional random field modeling [C]//Proc of Pacific Asia Conference on Language, Information and Computation. New York: ACM Press, 2006: 87-94.

[8] He Jia, Li Guanghong. Research of Chinese word segmentation based on neural network and particle swarm optimization [C]//Proc of the 3th International Conference on Apperceiving Computing and Intelligence Analysis. Piscataway, NJ: IEEE Press, 2010: 56-59.

[9] Zheng Xiaoqing, Chen Hanyang, Xu Tianyu. Deep learning for Chinese word segmentation and POS tagging [C]//Proc of the 18th Conference on Empirical Methods in Natural Language Processing. Stroudsbury, PA: ACL, 2013: 647-657.

[10] Chen Xinchi, Qiu Xipeng, Zhu Chenxi, et al. Gated recursive neural network for Chinese word segmentation [C]//Proc of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. Stroudsbury, PA: ACL, 2015: 1744-1753.

[11] Cai Deng, Zhao Hai, Zhang Zhisong, et al. Fast and accurate neural word segmentation for Chinese [C]//Proc of the 55th Annual Meeting of Association for Computational Linguistics. Stroudsbury, PA: ACL, 2017: 608-615.

[12] Han Dongxu, Chang Baobao. Domain adaptation method of Chinese word segmentation model [J]. Chinese Journal of Computers, 2015, 38(2): 272-281.

[13] Qiu Likun, Zhang Yue. Word segmentation for Chinese novels [C]//Proc of the 29th AAAI Conference on Artificial Intelligence. Menlo Park, CA: AAAI, 2015: 2440-2446.

[14] Zhang Shaoyang, Wang Peiyan, Cai Dongfeng. A multi-model of Chinese word segmentation based on character [J]. Journal of Shenyang Aerospace University, 2017, 34(1): 70-75.

[15] Ma Jianqiang, Hinrichs E. Accurate linear-time Chinese word segmentation via embedding matching [C]//Proc of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. Stroudsbury, PA: ACL, 2015: 1733-1743.

[16] Huang Changning, Zhao Hai. Ten years of Chinese participle review [J]. Journal of Chinese Information Processing, 2007, 21(3): 8-19.

[17] Sun Jigui, Liu Jie, Zhao Lianyu. Clustering algorithm research [J]. Journal of Software, 2008, 19(1): 48-61.

[18] Zheng Xiaoqing, Chen Hanyang, Xu Tianyu. Deep learning for Chinese word segmentation and POS tagging [C]//Proc of the 18th Conference on Empirical Methods in Natural Language Processing. Stroudsbury, PA: ACL, 2013: 647-657.

[19] Collobert R, Weston J, Bottou L, et al. Natural language processing (almost) from scratch [J]. Journal of Machine Learning Research, 2011, 12(1): 2493-2537.

[20] Pei Wenzhe, Ge Tao, Chang Baobao. Max-margin tensor neural network for Chinese word segmentation [C]//Proc of the 52th Annual Meeting of the Association for Computational Linguistics. Stroudsbury, PA: ACL, 2014: 293-303.

[21] Chen Xinchi, Qiu Xipeng, Zhu Chenxi, et al. Long short-term memory neural networks for Chinese word segmentation [C]//Proc of the 20th Conference on Empirical Methods in Natural Language Processing. Stroudsbury, PA: ACL, 2015: 1197-1206.

[22] Cai Deng, Zhao Hai. Neural word segmentation learning for Chinese [C]//Proc of the 54th Annual Meeting of the Association for Computational Linguistics. Stroudsbury, PA: ACL, 2016: 409-420.

[23] Song Yan, Cai Dongfeng, Zhang Guiping, et al. A Chinese word segmentation method based on joint decoding of words [J]. Journal of Software, 2009, 20(9): 2366-2375.

[24] Liu Jun, Ji Shuwang, Ye Jieping. Multi-task feature learning via efficient L2,1-norm minimization [C]//Proc of Conference on Uncertainty in Artificial Intelligence. 2009: 339-348.

[25] Chen Xinchi, Shi Zhan, Qiu Xipeng, et al. Adversarial multi-criteria learning for Chinese word segmentation [C]//Proc of the 55th Annual Meeting of the Association for Computational Linguistics. Stroudsbury, PA: ACL, 2017: 1193-1203.

[26] Liu Pengfei, Qiu Xipeng, Huang Xuanjing. Adversarial multi-task learning for text classification [C]//Proc of the 55th Annual Meeting of the Association for Computational Linguistics. Stroudsbury, PA: ACL, 2017: 1-10.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.