Available online at http://www.cnki.net

The Journal of China Universities of Posts and Telecommunications

April 2019, 26(2): 1-8

http://jcupt.bupt.edu.cn

Sentence segmentation for classical Chinese based on LSTM with radical embedding

Han Xu[1,2,3](✉), Wang Hongsu[4], Zhang Sanqian[5], Fu Qunchao[1,2], Liu Jun[5]

1. School of software Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
2. Key Laboratory of Trustworthy Distributed Computing and Service, (BUPT), Ministry of Education, Beijing 100876, China
3. The Key Laboratory of Rich-Media Knowledge Organization and Service of Digital Publishing Content, Institute of Scientific and Technical Information of China, Beijing 100038, China
4. Insitute of Quantitative Social Science, Harvard University, Cambridge, MA, USA
5. Department of statistics, Harvard University, Cambridge, MA, USA

Abstract

A low-than character feature embedding called radical embedding is proposed, and applied on a long-short term memory (LSTM) model for sentence segmentation of pre-modern Chinese texts. The dataset includes over 150 classical Chinese books from 3 different dynasties and contains different literary styles. LSTM-conditional random fields (LSTM-CRF) model is a state-of-the-art method for the sequence labeling problem. This model adds a component of radical embedding, which leads to improved performances. Experimental results based on the aforementioned Chinese books demonstrate better accuracy than earlier methods on sentence segmentation, especial in Tang's epitaph texts (achieving an F_1-score of 81.34%).

Keywords LSTM, radical embedding, sentence segmentation

1 Introduction

Many Asian languages including Chinese do not put space between characters. It is clear that word segmentation is one of the first and foremost tasks in natural language processing (NLP) for these languages. Throughout the years, researchers have made significant processes in this task [1-4]. However, state-of-the-art techniques for segmentation of Chinese texts have almost exclusively focused on modern Chinese.

Pre-modern Chinese, or classical Chinese, refers to recorded Chinese texts from 1 600 B.C to 1 800 A.D. These texts contain rich information on Chinese language, literature and history. With rapid development of optical character recognition (OCR) techniques, many classical Chinese books have been digitalized to texts. For example, the largest commercial text and image depository in China, Ding-Xiu full text search platform, contains 6 billion characters. The largest noncommercial text and image depository, the Chinese text project, includes 5 billion characters. This gives rise to exciting opportunities in using computational techniques to retrieve and analyze

Received date: 07-08-2018
Corresponding author: Han Xu, E-mail: fionaxuhan@gmail.com
DOI: 10.19682/j.cnki.1005-8885.2019.1001

these texts. For example, the Chinese biographical database project (Harvard University, Peking University 2018. China biographical database (CBDB) https://projects.iq.harvard.edu/cbdb) systematically extracts data from historical texts and converts them into a database using regular expression and tagging. However, even as OCR techniques mature, there are still obstacles standing between standard computational methods and texts retrieved from OCR.

First, Chinese texts have no spacing between characters or words. Second, the grammar of classical Chinese is significantly different from modern Chinese. Hence, commonly used corpus based on modern Chinese is not useful in processing classical Chinese texts. Third, individual characters have a richer set of meanings in classical Chinese than in modern ones, which makes it more ambiguous to define 'words' from combinations of characters linguistically. As a result, the word segmentation task for classical Chinese texts is more difficult and less well defined than that for modern Chinese. Fourth, classical Chinese texts seen in historical records have no punctuations at all, whereas [5] argued that punctuation has very important meanings in NLP, especially in Chinese. Fig. 1 is an example of classical Chinese. This introduces a severe problem of sentence segmentation (in addition to word segmentation) for processing raw classical Chinese texts from OCR.

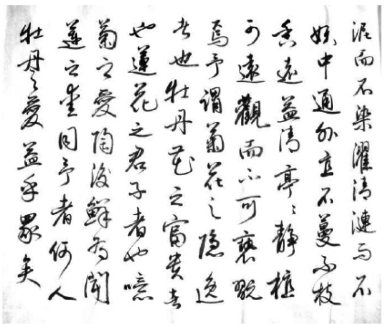

Fig. 1 A classical Chinese texts example: it is a famous prose from dynasty Song

We propose a LSTM model with radical embedding to solve the pre-modern Chinese sentence segmentation problem. The main contributions of this work include that: 1) We develop an algorithm based on the Bidirectional-LSTM-CRF model with radical embedding; 2) We provide a pre-modern character embedding in a huge corpus; 3) We train a model for each dynasty separately, which results in a higher accuracy than training one model for all the text data.

2 Related work

Sentence boundary detection (SBD) is a language problem. The main purpose is to identify suitable breaks in sentences. Most existing research in SBD focus on the problem in the context of analyzing speech. There has not been much work in SBD for written text. The key of these problems is parts-of-speech tagging. However, for pre-modern Chinese, there's no perfect corpus withpart-of-speech (POS) tagging for now, and it's hard to define POS in pre-modern Chinese, so it's more difficult to solve this problem.

There have been a couple of works focusing on SBD for classical Chinese in the Chinese research community. Ref. [6] proposed an algorithm to break sentences for ancient Chinese agricultural books using regular expression. They identified special syntax words and introduced punctuation around these syntax words. Ref. [7] are the first in using modern computational models for sentence segmentation in classical Chinese. Their algorithm is based on a n-gram model. Ref. [8] used a cascaded CRF model, which achieved better results than Ref. [7]. Ref. [9] built upon CRF-based models by integrating phonetic information in Chinese. However, such information heavily depends on professional inputs. More recently, Ref. [10] used the recurrent neural network (RNN) model for sentence segmentation which is similar to our setup.

In 1986, Rumelhar and Hinton proposed the back propagation algorithm [11]. After that, researchers developed various of convolutional neural network (CNN) and RNN to solve NLP tasks. Bengio tried to use neural network to build language model in 2003 [12]. In 1997, Ref. [13] first proposed the LSTM model, and gave the details about the main structures. In 2014, Ref. [14] changed the calculation of output gate. They used tanh function instead of sigmoid

function, and achieved better performances. Ref. [15] proposed two regularization methods to solve the vanishing gradient and exploding gradient problems in RNN. Ref. [16] proposed the Bi-LSTM-CRF model, which is almost state-of-the-art. It added CRF as final layer. For now, the most popular method is CNN-Bi-LSTM-CRF [17]. However, they regard English letters as pixels, while in Chinese, each character is single and independent and it is hard to split.

Ref. [18] used Bi-LSTM-CRF to do named entity recognition (NER) task. They also used character embedding to improve the performance. In English, each word is formed by many letters, as there is root, prefix and suffix in English words which has similar meanings in some occasions, so embedding by letters may get better results. However, in Chinese, every character is single and independent, so character embedding doesn't suit well. Ref. [19] proposed to use radical information to improve Chinese embedding. It's similar to the character embedding in English. But the results only slightly better than baseline models.

We use Bi-LSTM-CRF model with radical embedding. Compared to work by Ref. [10], our paper uses state-of-the-art model with CRF. Moreover, we add radical embedding as the input, the results are better than Ref. [10].

3 Model

3.1 Radical embedding

Ref. [20] proved that word embedding could highly improve performances in sequence tagging problems. In our model, we use vector representation of individual character as the input. In Chinese, most characters have a radical, which is analogous to prefix and suffix in English. It is often the case that characters that share the same radical have related meanings. As shown in Fig. 2, radical '月' (moon, 'mutated' from the character for meat) usually means body parts as a radical, such as '腿' (leg, unicode: u817f), '膊' (arm, unicode: u818a). Some simplified Chinese characters lost some radicals. For example, the upper radical of the character '雲' (cloud, unicode: u96f2) is '雨', which means rain; whereas the simplified one '云' only keeps the bottom part. Hence, radical embedding may capture more information in pre-modern Chinese by a corresponding segmentation method than in modern Chinese.

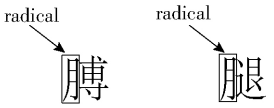

Fig. 2 Character radical example

Every Chinese character can be represented by a unique Unicode. Characters that share the same radical are grouped together in Unicode, which can be easily retrieved in our model. The authoritative Xinhua Dictionary [21] reveals 214 radicals in total. When generating character embedding, we take these 214 radicals as parts of the input, represent them initially by randomly generated vectors, and train together as parts of the word representation. We use a continuous bag-of-words (CBOW) model for radicals, which is similar to the original CBOW model [22]. We modify the model such that each character is represented by a concatenated vector based on the character and its radical part, and maximize the log-likelihood function.

$$L = \sum_{x_i^a} \mathrm{lb}P(x_i \mid h_i) \tag{1}$$

where x_i is the output Chinese character, h_i is the concatenation of c_i and r_i.

$$h_i = \mathrm{cat}(c_{i-N}, r_{i-N}, \ldots, c_{i+N}, r_{i+N}) \tag{2}$$

where c_i is the character part and r_i is the radical part, N is the window size.

We make prediction using a matrix $\boldsymbol{W}$. Different from the original CBOW model, the extra parameter introduced in the matrix $\boldsymbol{W}$ allows us to maintain the relative order of the components and treat the radical differently from the rest components.

3.2 Bidirectional LSTM

RNNs are one kind of neural networks that deal with sequential problems. In theory, RNN can handle the long-term dependencies task, but actually, it is hard to learn well when input information is too long. Also, it cannot choose important previous information. It just calculates all information without being weighted. Our model follows Ref. [17]'s work, which used Bi-LSTM

to solve sequence problems. LSTM includes the memory-cell in its hidden layer, which is the key of this model. The model has 3 gates: input, forget, output gate. It controls proportion of information taken from the input, the proportion of previous information to forget, and feed information to next time step. All calculations are done on the cell state. The cell receives 2 parameters h_{t-1} and c_{t-1} from the previous time step -1, and input x_t from the current time step. Fig. 3 shows the structure of a LSTM at time step t.

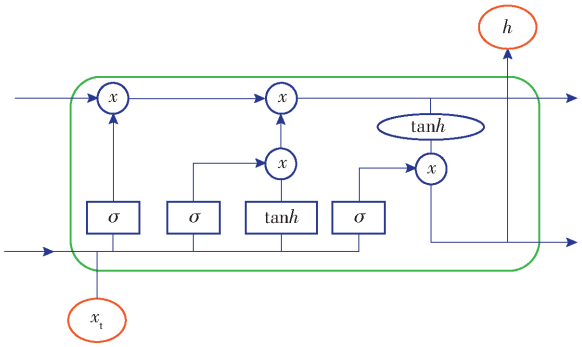

Fig. 3 Structure of LSTM

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (3)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{i-1} + W_{cf}c_{t-1} + b_j) \quad (4)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (5)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (6)$$

$$h_t = o_t \tanh(c_t) \quad (7)$$

where i_t, f_t and o_t are the input, forget, and output gates, respectively. c_t represents the cell state and h_t denotes the hidden layer parameter at the current time step. All of these have the same dimensionality as the size of the hidden layer. σ is a standard sigmoid function, the W's are the matrices, and the b's are the bias terms.

3.3 CRF layer

In our model, we replace the softmax layer by a CRF layer. As shown in Fig. 4, in the CRF layer, the adjacent outputs are linked each other, so we can get an optimal tagging sequence instead of an independent tagging.

$$s(x, y) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=0}^{n} p_{i, y_i} \quad (8)$$

The input matrix P is of size $n \times k$, where k is the number of tagging. $\boldsymbol{A}$ is a state transition matrix, where A_{ij} represents the transition probability from state i to state j. The optimal $s(x, y)$ can be obtained by dynamic programming.

In our model, the input is a character sequence $(x_1, x_2, \ldots, x_n)$, and the output is a vector $(y_1, y_2, \ldots, y_n)$, where each y_i is a probability vector corresponding to each tagging. Fig. 4 shows the detail of our algorithm.

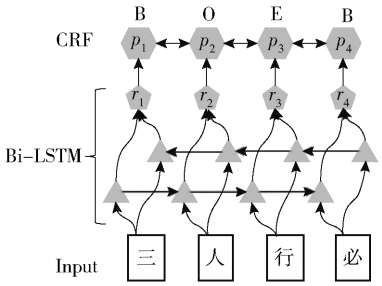

Fig. 4 Algorithm details

We use the Bi-LSTM-CRF model to carry out sentence breaking operations for ancient Chinese. This model can capture the context information of sentences and enhance the semantic relevance. By adding CRF layer, the output layer of the model builds certain logic rules. Compared with LSTM, the optimal path can be found, instead of the maximum probability of each output.

4 Experiments

4.1 Data set

We obtain 150 ancient texts from CBDB. The total number of characters is 44 083 978, and the vocabulary size is 20 285. Table 1 shows the dataset details.

Table 1 Dataset details

Dynasty	Total characters	Vocabulary size
Tang	6 160 233	7 478
Ming	7 414 125	9 952
Qing	29 392 770	1 0151
Tang-epitaph	1 116 850	5 197

We divide out texts according to dynasties and literary styles: dynasty Tang (A. D. 618 — 907), dynasty Ming (A. D. 1 368 — 1 644), dynasty Qing

(A. D. 1 644 — 1 912), and Tang's epitaph. As most of the epitaphs are engraved on stones, some of the characters cannot be recognized due to corrosions caused by harsh weathers. Thus, there are a lot of unsure characters indicated by '□', which is difficult to tag. To ensure sensible outputs, we deleted sentences that have more than 5 consecutive '□'.

4.2 Tagging schemes

The most popular tagging method in NER tasks usesbegin-inside-otherwise (BIO) format. B indicates the beginning of an entity, I means that the character is inside an entity, and O means otherwise. We found that both of the beginning and the end of a sentence have significant feature. Inspired by BIO tags, we assign E as the end of a split sentence, B as the beginning of a split sentence, other characters are tagged by O. This tagging method is simple, and we only place punctuation marks between E and B to make sure the accuracy. Fig. 5 shows a tagging result of a famous proverb of confucius.

B	O	E		B	O	O	O	E	
三	人	行	,	必	有	我	师	焉	。

Fig. 5 Tagging example

This is a well-known sentence from confucius. Translation: there is always someone, among any three people, who can teach me.

4.3 Training details

We take over 150 books texts as the training dataset, and obtain a classical Chinese embedding. First, we delete most punctuation marks and only keep those that mean full and half stops in the sentences. In this experiment, the punctuation set we kept is {; : , . ? !}. Then, we separate the texts into units by dynasty. Each unit contains 100 characters, about 6 ~ 7 sentences. These units are broken up first, and split into a training set, a validation set, and a test set. The proportions are 50%, 25%, and 25%, respectively. These three sets have their own purposes. Training set is for train our model, to find the best $\boldsymbol{W}$ and b; validation set can evaluate whether it's a better model; while test set is only for final performance evaluation which only be used once. It is necessary to ensure that the test set is completely independent. The test set should be sealed until the model adjustment and parameter training are completed. We followed the Ref. [20]'s work of the hyper parameters, Table 2 shows the details of the parameters. Each dynasty and literary style have one model, so we totally get 4 models, which is dynasty Tang, dynasty Ming, dynasty Qing and epitaph for dynasty Tang.

Table 2 Experiment hyper parameters

Hyper	parameters
Word embedding dimension	100
Hidden layer size	100
Hidden layer	1
Batch	50
Epoch	30
Learning rate	0.01
Gradient clipping	5
Dropout rate	0.5

The whole process is as follows: First, we use training set to train the model and find the optimal function to minimize the loss function. Each time the model randomly picks up one unit in training set and begin training, when it updates the $\boldsymbol{W}$ and b, it drops off this unit and randomly picks up other one. Second, we measure the optimal function on the verification set. In own model, we measure 10 times in one epoch. We repeat these two steps until the epochs is done. After that, the model with minimum error on the verification set is selected, and the training set and verification set are combined as the whole training model to find the optimal function. Finally, the generalization performance of the optimal function is measured on the test set. We use Precision, Recall, and F_1 to evaluate our model.

In this method, the values of true positive (TP), false positive (FP), true negative (TN), false negative (FN) are all focus on the accuracy of stop signs, not the accuracy of character tags.

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{9}$$

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{10}$$

$$F_1 = \frac{2PR}{P+R} \tag{11}$$

4.4 Results

Tables 3 – 6 show the results on different datasets. It is seen that our new method performs the best according to the F_1 score in all datasets (arranged by dynasty and literary style). The results show that the Bi-LSTM-CRF model performs better than that LSTM. This is because the two-way network can not only capture the forward and backward information, but can also build certain logical rules to restrict the label output. In addition, the radical embedding method also has a good performance. This proves that the study of Chinese hieroglyphics can improve the overall effect of ancient Chinese. All methods perform the best for Tang's epitaph. It proves that in additional to classify by dynasty, classify by literary styles can be more reasonable. Our model performs the next best for the texts from the dynasty Qing. As the Qing has the largest dataset, and its period is the closest to modern times, the good performance of our algorithm (and CRF) may be due to both a better training and a better definition of sentence structures (according to the modern Chinese rules) of the texts.

Table 3 Sentence segmentation results for Tang

Tang	P	R	F_1-measure
CRF	0.757	0.715	0.735
LSTM	0.766	0.696	0.729
Bi-LSTM-CRF	0.734	0.728	0.731
Bi-LSTM-CRF with radical embedding	0.747	0.748	0.748

Table 4 Sentence segmentation results for Ming

Ming	P	R	F_1-measure
CRF	0.624	0.767	0.689
LSTM	0.669	0.672	0.671
Bi-LSTM-CRF	0.722	0.675	0.698
Bi-LSTM-CRF with radical embedding	0.696	0.714	0.705

Table 5 Sentence segmentation results for Qing

Qing	P	R	F_1-measure
CRF	0.779	0.759	0.769
LSTM	0.686	0.742	0.713
Bi-LSTM-CRF	0.741	0.765	0.752
Bi-LSTM-CRF with radical embedding	0.761	0.784	0.772

Table 6 Sentence segmentation results for Tang epitaph

Tang-epitaph	P	R	F_1-measure
CRF	0.803	0.795	0.799
LSTM	0.759	0.781	0.771
Bi-LSTM-CRF	0.789	0.826	0.807
Bi-LSTM-CRF with radical embedding	0.814	0.813	0.813

Taking the epitaph of thedynasty Tang as an example, Fig. 6 shows the P, R and F_1 value of the model. We can see that the convergence speed of the model in the early stage is fast. At the 40 times, the model has begun to stabilize, and when it came to about 80 times, the model has basically reached the optimal value.

We also train embedding by different dynasties.

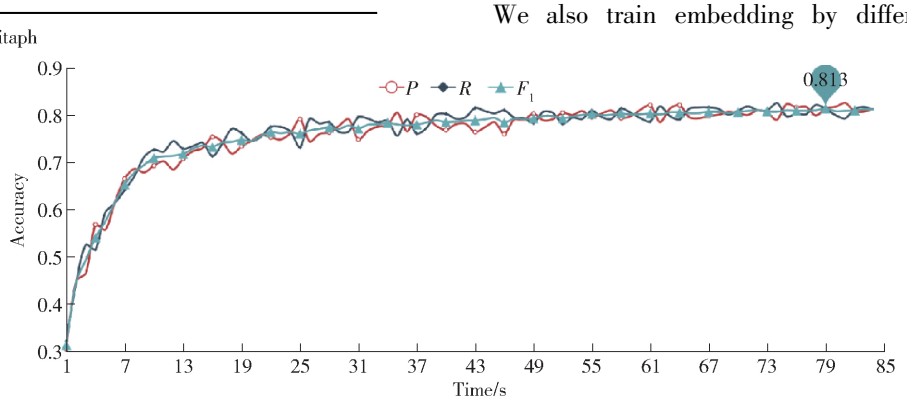

Fig. 6 P, R, F_1 value of Tang epitaph

Fig. 7 shows the results of Tang's epitaph in different embedding method. As we can see, a unique embedding performs better than general embedding. This may be because that the language style of epitaph is different from other texts and is easier and more 'standardized'.

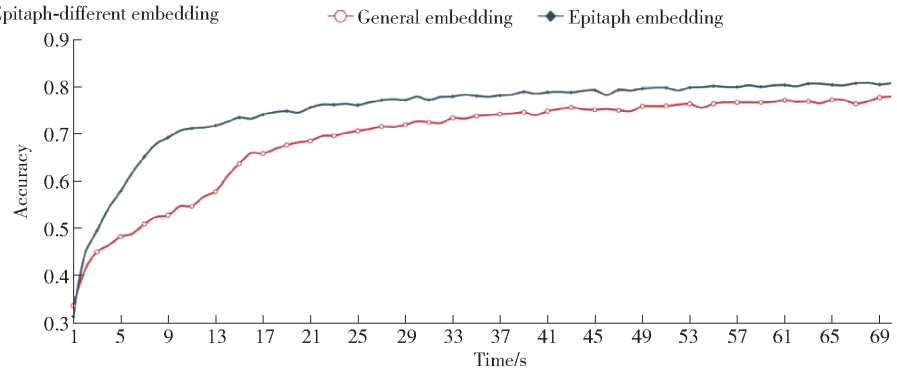

Fig. 7 Results of general embedding and epitaph embedding on epitaph dataset

Fig. 8 shows the example of a result of dynasty Qing, it is part of a biography about a famous poet called Dongpo hermit. Comparing with the original text, we can see that the model marked most of the correct position of sentence breaks, with a high TP. Otherwise, the model is more inclined to mark sentences with short sentences, and the position of the clause can be explained.

Our model:	**先生**(Prof. su) /**年四十七**(is 47 years old) /**在黄州**(in Province Huang) /**寓居临皋亭**(lives in pavilion Linao) / **就东坡**(near Dongpo(Location)) /**筑雪堂**(builds a house called Xuetang) /**自号东坡居士**(called Dongpo hermit by himself) /**以东坡图考之**(based on the map of Dongpo) /**自黄州门南**(from the south door of Province Huang) /**至雪堂四百三十步**(430 feet to Xuetang)
Orginal:	**先生年四十七**(Prof. Su is 47 years old) /**在黄州**(in Province Huang) /**寓居临皋亭**(lives in pavilion Linao) / **就东坡筑雪堂**(builds a house near Dongpo(Location) called Xuetang) /**自号东坡居士**(called Dongpo hermit by himself) /**以东坡图考之**(based on the map of Dongpo) /**自黄州门南至雪堂四百三十步**(Xuetang is 430 feet from the south door of province Huang)

Fig. 8 Case studies

According to the demonstration and analysis of the experimental results, it can be basically proved that our model is effective in ancient Chinese sentence segmentation.

5 Conclusions

A modified Bi-LSTM-CRF model with radical embedding is proposed. Our experiments show that this model outperforms existing methods in all the pre-modern Chinese text datasets we tested. The key of this new model is that we first conduct word embedding for the radicals together with the corresponding characters in the pre-training, and then use this joint embedding as the input parameter. While some earlier studies showed that including radicals do not seem to help segmenting modern Chinese texts, our study demonstrates that radicals can provide us a better handle on classical Chinese. This is consistent with the common wisdom that the 'shape' of a character is more important and meaningful in pre-modern Chinese than in modern Chinese. In the future work, we hope to not only break sentences, but also tag the punctuation appropriately, which is desirable but much more challenging.

Acknowledgements

This work was supported by the Fund of the key laboratory of rich-media knowledge organization and service of digital publishing content (ZD2018-07/05) .

References

1. Xue N, Shen L. Chinese word segmentation as LMR tagging. Proceedings of the Second SIGHAN Workshop on Chinese Language Processing-Volume 17. Association for Computational Linguistics, 2003: 176 – 179
2. Peng F, Feng F, McCallum A. Chinese segmentation and new word detection using conditional random fields. Proceedings of the 20th International Conference on Computational Linguistics. Association for Computational Linguistics, 2004: 562 – 568
3. Chen X, Qiu X, Zhu C, et al. Long short-term memory neural networks for Chinese word segmentation. Conference on Empirical Methods in Natural Language Processing, 2015: 1197 – 1206
4. Zheng X, Chen H, Xu T. Deep learning for Chinese word segmentation and POS tagging. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013: 647 – 657
5. Li Z, Sun M. Punctuation as implicit annotations for Chinese word segmentation. Computational Linguistics, 2009, 35(4): 505 – 512
6. Huang J, Hou H. A research on punctuation pattern of ancient agricultural text. Journal of Chinese Information, 2008, 22(4): 31 – 38 (in Chinese)
7. Chen T, Chen R, Pan L, et al. Sentence segmentation in ancient Chinese based on N-gram model. Computer Engineering, 2007, 33(3): 192 – 193 (in Chinese)
8. Zhang K, Xia Y, Yu H. An ancient Chinese punctuation and sentence marking method based on cascaded CRF. Computer Application Research, 2009(10): 40p (in Chinese)
9. Huang H H, Sun C T, Chen H H. Classical Chinese sentence segmentation. CIPS-SIGHAN Joint Conference on Chinese Language Processing, 2010
10. Wang B, Shi X, Tan Z, et al. A sentence segmentation method for ancient Chinese texts based on NNLM. The Workshop on Chinese Lexical Semantics. Springer International Publishing, 2016: 387 – 396
11. Rumelhart D E, Hinton G E, Williams R J. Learning internal representations by error propagation. California Univ San Diego La Jolla Inst for Cognitive Science, 1985
12. Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model. Journal of Machine Learning Research, 2003, 3(2): 1137 – 1155
13. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9(8): 1735 – 1780
14. Jozefowicz R, Zaremba W, Sutskever I. An empirical exploration of recurrent network architectures. International Conference on Machine Learning, 2015: 2342 – 2350
15. Pascanu R, Mikolov T, Bengio Y. On the difficulty of training recurrent neural networks. International Conference on Machine Learning, 2013: 1310 – 1318
16. Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging. ArXiv Preprint ArXiv: 1508.01991, 2015
17. Ma X, Hovy E. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. ArXiv Preprint ArXiv: 1603.01354, 2016
18. Lample G, Ballesteros M, Subramanian S, et al. Neural architectures for named entity recognition. ArXiv Preprint ArXiv: 1603.01360, 2016
19. Shao Y, Hardmeier C, Tiedemann J, et al. Character-based joint segmentation and POS tagging for Chinese using bidirectional RNN-CRF. ArXiv Preprint ArXiv: 1704.01314, 2017
20. Collobert R, Weston J, Bottou L, et al. Natural language processing (almost) from scratch. Journal of Machine Learning Research, 2011, 12(8): 2493 – 2537
21. Compilation of Xinhua Dictionary. Xinhua dictionary. 10th edition. Commercial Press, 2004 (in Chinese)
22. Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space. ArXiv Preprint ArXiv: 1301.3781, 2013

(Editor: Ai Lisha)