New Word Detection in Ancient Chinese Literature

Tao Xie, Bin Wu, and Bai Wang

Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia, School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China 100876
xietao0222@bupt.edu.cn, wubin@bupt.edu.cn, wangbai@bupt.edu.cn

Abstract. Mining Ancient Chinese corpus is not as convenient as modern Chinese, because there is no complete dictionary of ancient Chinese words which leads to the bad performance of tokenizers. So finding new words in ancient Chinese texts is significant. In this paper, the Apriori algorithm is improved and used to produce candidate character sequences. And a long short-term memory (LSTM) neural network is used to identify the boundaries of the word. Furthermore, we design word confidence feature to measure the confidence score of new words. The experimental results demonstrate that the improved Apriori-like algorithm can greatly improve the recall rate of valid candidate character sequences, and the average accuracy of our method on new word detection raise to 89.7%.

Keywords: New word detection, Ancient Chinese literature, Apriori-like, Neural network, Word confidence

1 Introduction

Detecting new words in corpus has great significance in natural language processing (NLP), and it is indispensable to word segmentation, named entity recognition and other tasks. According to study of Ma and Chen (2003) [4], more than 62% of word segmentation errors derive from word out of dictionary.

Nowadays, Chinese new word detection mostly focuses on modern Chinese corpus. The research in ancient Chinese corpus is very limited. However, ancient Chinese is quite different from modern Chinese in several ways, such as words, phrases and syntactic structure. Generally, technique used in modern Chinese word detection may not be suitable for ancient Chinese. Moreover, many words used today, cannot be treated as a word in ancient Chinese. For example, the word '可以' in modern Chinese means 'can'. Meanwhile, in ancient Chinese, it has to be treated as two separate words, when these two words occur together, they mean 'can rely on'.

Deng et al (2016) [7] has proposed an unsupervised method for simultaneously discovering and segmenting words and phrases from domain-specific Chinese texts. Although it was useful for segmenting ancient Chinese texts, the granularity of word segmentation was uneven. There were a lot of segmentation ambiguities, and the effect of mining low frequency words was limited. However, a

lot of new words in the ancient Chinese literature belong to low frequency words and there are few segmentation ambiguities in the ancient Chinese words. For this reason, a novel model was proposed to detect new words of ancient Chinese literature in this paper.

This paper puts forward a novel model using improved Apriori-like algorithm and LSTM neural network to mine new words in ancient Chinese texts. Apriori-like algorithm was used to generate candidate character sequences. Traditional Apriori algorithm can hardly find low frequency new words and may produce a large amount of noise words. We have improved the original Apriori algorithm and proposed the rule of word formation for low frequency words, which increased the recall rate of valid candidate character sequences greatly.

Recently, neural network models have increasingly been used for NLP tasks for their ability to minimize the effort in feature engineering. Long short-term memory (LSTM) neural network was used to identify the boundaries of candidate character sequences in this paper. We use LSTM network to acquire segmentation probability between two characters. Then, candidate character sequences were classified to new words and noise words based on their segmentation probability sequences.

Our major contributions include:

1. We propose a novel model called AP-LSTM, which combining Apriori-like algorithm and LSTM neural network together. This model improves the accuracy of detecting new words of ancient Chinese literature greatly.
2. Improved Apriori-like algorithm breaks the bottleneck of identifying low frequency new words, and greatly reduces the redundant items of candidate character sequences.
3. We propose word confidence feature to measure the probability score of new words, which indicates how likely the new word would be a valid word.

The rest of the paper is organized as follows. Section 2 discusses related research. Section 3 gives an overview of AP-LSTM model. Details of improved Apriori-like algorithm and LSTM neural network model are discussed in Section 4 and 5. Section 6 shows the filtering rule of new words and word confidence. Experiments and results are described in Section 7, and Section 8 summarizes the conclusion and future works of our method.

2 Relevant Work

Generally speaking, Chinese new word detection interweaves with Chinese word segmentation, particularly in Chinese NLP. In these works, new word detection is considered as an integral part of segmentation, where new words are identified as the most probable segments inferred by the probabilistic models. Typically models include conditional random fields proposed by Peng F et al. (2004) [12], and a combined model trained with adaptive online gradient descent based on feature frequency information (Sun et al. 2012 [13]).

Another line is regarding new word detection as a separate task. The first genre of such studies is to employ complex linguistic rules or knowledge. For

example, Justeson and Katz (1995) [11] extracted technical terminologies from documents using a regular expression. Chen and Ma (2002) [5] have combined morphological and statistical rules to detect Chinese new word. These methods require engineering of linguistic features and their scalability is poor. The second genre of the studies is to use statistical methods and regarded new word detection as multi-word expression extraction. The first model for quantifying multi-word association is Pointwise Mutual Information (PMI) (Chruch and Hanks, 1990) [6]. Zhang et al. (2009) [16] has proposed Enhanced Mutual Information (EMI) which measures the cohesion of n-gram using the frequency of its own and the frequency of each sub-word. Bu et al. (2010) [3] has proposed a new feature named multi-word expression distance (MED). However, the capacity of these statistical methods to detect low frequency words is limited. And multi-features fusion was used frequently to improve the precision and reliability of the recognition.

In addition, user behavior data has recently been explored to find new word. Zheng et al. (2009) [18] has utilized user typing custom in Sogou Chinese Pinyin input method to detect new words. Zhang et al. (2010) [17] has used dynamic time warping to detect new words from query logs. These works performed well, however, they were restrained by the unavailability of expensive commercial resources.

Zhang H et al. (2014) [15] has proposed a pragmatic quantitative model to analyze and estimate the performance of new word detection. Huang m et al. (2014) [10] has proposed statistical measures to quantify the utility of a lexical pattern and detect new sentiment words. Their works heavily focused on evaluation of new word detection.

All of these works above mainly focused on modern Chinese. In ancient Chinese, the performance of their methods is limited to the in-depth linguistic knowledge and widely distribution of low frequency words.

3 Problem Statement

Before discussing about the model proposed in this paper, we need to figure out which words belong to new words in ancient Chinese and identify our problem. As for "new word", it is a word with specific meaning firstly. Second, they occur very rarely in modern Chinese corpus. Finally, they are characterized with the ancient era and various regions, which involved a classical Chinese words, poetry vocabulary, terminology and so on. So the description of new words in ancient Chinese is as follow:

New words in ancient Chinese are sequences of characters which contain a clear semantic interpretation and the historical characteristics. Meanwhile they are not included in the standard dictionary.

Based on the above definition, we can know the meaning of new words in ancient Chinese. And we designed a standard modern Chinese lexicon (used in rest paper) to filter out the noise words, including modern Chinese and stop words. Furtherly, we can present the new word detection problem formally as follows:

Definition 1. *For an given ancient Chinese corpus $\mathcal{C}$ and a standard dictionary $\mathcal{D}$. New word detection problem aims at finding all new words $\mathcal{W}$ which are out of $\mathcal{D}$. It consists of generating candidate new words $\mathcal{A}$ and the filtering of noise new words $\mathcal{T}$. In other words, new word set $(\mathcal{A} - \mathcal{T})$ is the final result.*

4 An Overview of AP-LSTM Model

The AP-LSTM model consists of two steps: generation step and selection step. Generation step is used to produce candidate new words in the original corpus. Noise new words are filtered out during selection step. We utilized improved Apriori-like algorithm to generate candidate character sequences and used LSTM neural network to recognize the boundaries of words. Finally, we pruned the candidate new words based on the filtering rule. In addition, word confidence could be used to measure the score of a candidate new word being a valid word. New words can be further mined based on this feature. The procedure of AP-LSTM is as follows:

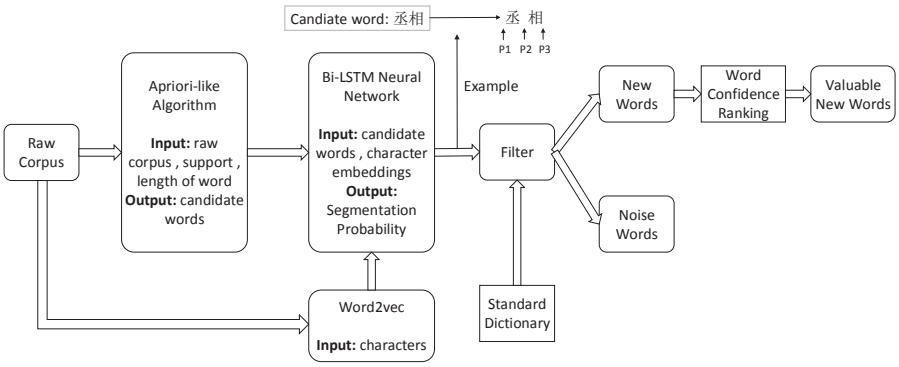


Fig. 1: Flow of AP-LSTM

For example, the word '丞相' (prime minister) generated by Apriori-like algorithm will acquire three segmentation probabilities computed by LSTM. Then, we classified it to new word or noise word based on filtering rule and standard dictionary. Finally, word confidence score is measured by its segmentation probabilities. And we can mine more valuable new words based on word confidence score ranking.

5 Improved Apriori-like Algorithm

Traditional Apriori algorithm was applied in association rule and frequent itemsets mining. Since it was proposed by Agrawal R et al. [1] in 1994. It not only influenced the association rule mining community, but also affected other data mining fields. In recent years, some researchers have applied Apriori algorithm

to NLP tasks and used it to generate domain-specific words. Besides, Wang et al. (2006) [14] has used Apriori algorithm to generate frequent items, then filter frequent items by confidence and acquire specific words. However, this method will generate a lot of noise words and redundant items.

In this paper, we improved Apriori algorithm from two aspects and used it to generate candidate new words $\mathcal{A}$:

1. Candidate generation.
2. Finding low frequency new words.

5.1 Candidate Generation

Original Apriori algorithm contains the join step and the pruning step. Join opera-tion is used to generate candidate items, as shown in table 1.

Table 1: Candidate generation in Apriori

Join L_{k-1} **with** L_{k-1}
select $P\,(p_1p_2\cdots p_{k-1})$ **and** $Q\,(q_1q_2\cdots q_{k-1})$ **from** L_{k-1}
if $p_1p_2\cdots p_{k-2} = q_1q_2\cdots q_{k-2}$, $p_{k-1} \neq q_{k-1}$
insert $p_1p_2\cdots p_{k-1}q_{k-1}$ **into** C_k

When Apriori algorithm is applied to Chinese new word detection. In the table 1, L_{k-1} denotes candidate words of length k-1, p_1, q_1, ..., p_{k-1}, q_{k-1} are characters in candidate word P and Q. C_k denotes candidate character sequences of length k. Although this method can mining frequency candidate character sequences, it will generate a large amount of noise words and will not take the order of characters into consideration. Therefore, we improved original join operation, as shown in table 2. And a specific example of join operation between original Apriori algorithm and our Apriori-like algorithm is shown in figure 2.

Table 2: Candidate generation in new Apriori

Join L_{k-1} **with** L_{k-1}
select $P\,(p_1p_2\cdots p_{k-1})$ **and** $Q\,(q_1q_2\cdots q_{k-1})$ **from** L_{k-1}
if $p_2p_3\cdots p_{k-1} = q_1q_2\cdots q_{k-2}$
insert $p_1p_2\cdots p_{k-1}q_{k-1}$ **into** C_k

In the Apriori-like algorithm, we presumed that the character sequences is ordered. It means every non-empty subsequence of frequent itemsets is ordered. The improved Apriori algorithm do not need pruning step, According to the two properties of Apriori:

1. All the non-empty subset of frequent itemsets are frequent itemsets.

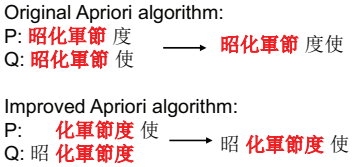
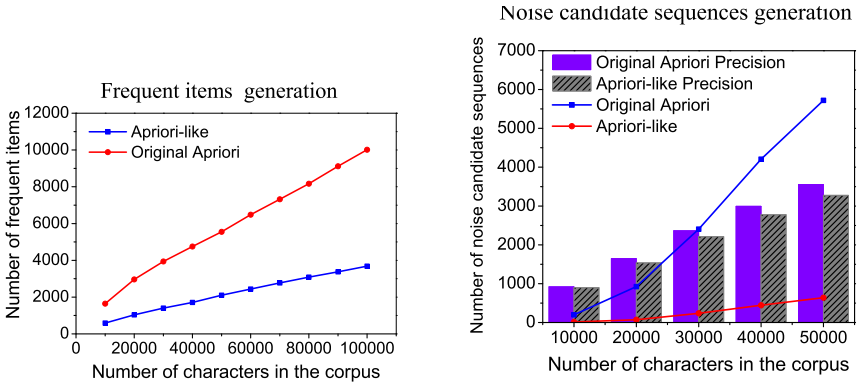


Fig. 2: Example of join operation



(a) (b)

Fig. 3: Histogram denotes the number of valid words. (The experiment corpus used here is Song Poetry.)

2. All the superset of non-frequent itemsets must be non-frequent itemsets.

Obviously, candidate frequent itemsets generated by new join operation can satisfy the two properties mentioned above. For example, k-frequent item p only has two non-empty subsequences ("$p_1p_2\cdots p_{k-1}$" and "$p_2p_3\cdots p_k$"), which are both coming from (k-1)-frequent itemsets. So Apriori-like algorithm can greatly reduce the noise words and take the order of characters into account. Figure 3 shows the comparison results of two Apriori algorithm.

Figure 3(a) shows that the number of frequent items generated by Apriori-like algorithm has been greatly reduced compared with the original Apriori algorithm. And experiment demonstrated that the Apriori-like algorithm can eliminate a lot of noise new words, almost without any impact on accuracy. Result is shown in Figure 3(b). In addition, this method is more accordant with word formation rule of characters.

5.2 Finding Low Frequency New Word

Although new Apriori-like algorithm can find a lot of valid candidate new words (Here valid new words denotes word with specific semantics), there are still plenty of new words could not been mined by algorithm. Then we found that a big part of words in ancient Chinese Literature only occur once or twice in the whole corpus. Frequent itemsets generation is based on support, thus new

Apriori-like algorithm cannot find low frequency new words of ancient Chinese literature. For this reason, we defined word formation rule for low frequency words.

Rule 1: *Low frequency character sequence made of frequent itemsets is more likely to be a new word.*

Based on this rule, we further improved Apriori algorithm. The algorithm flow is as follows.

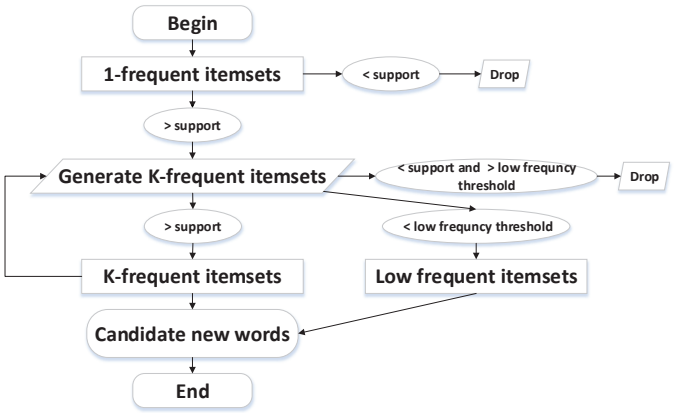


Fig. 4: Algorithm flow of Apriori-like algorithm

As shown in Figure 4, low frequency threshold was added to the algorithm. At every step of the iteration, k-frequent item that support less than threshold was found out and added into low frequency itemsets. Finally low frequency itemsets would also be added to candidate new words. In this paper, we set the threshold to 2. Later our experiment demonstrated that the Apriori-like algorithm can increase recall rate of valid candidate character sequence in ancient Chinese literature greatly.

6 Long Short-Term Memory Neural Network Model

Traditional new word detection usually artificially constructs word features to classify candidate character sequences. However, these features often cannot comprehensively acquire information of word. But methods of neural network and Embedding can address the bottleneck of feature engineering. LSTM neural network (Graves et al. [9]) is an extension of the recurrent neural network (RNN). Since LSTM neural can keep the previous import information in memory cell and avoid the limit of window size of local context, it is widely applied to NLP tasks. In this paper, LSTM neural network was used to identify the boundary of candidate new words.

6.1 Character Embeddings

The first step of processing symbolic data using neural network is representing them into distributed vectors, namely embeddings (Bengio et al. [2]). Formally, in new word detection task, we have a character dictionary C of size $|C|$. Each character $c \in C$ is represented as a real-valued vector (character embeddings) $v_{\mathrm{c}} \in R^d$ where d is the dimensionality of the vector space. The character embeddings are then stacked into an embedding matrix $M \in R^{d \times |C|}$. For a character $c \in C$, the corresponding character embedding $v_{\mathrm{c}} \in R^d$ is retrieved by the lookup table layer.

6.2 Segmentation Probability Model

We regard acquiring probability between two characters as character-based sequence labeling problem. Each character context is labeled as one of 1, 0 to indicate the segmentation. 1 represents segmented and 0 represents non-segmented, where size of context is even. So segmentation probability denotes the probability of being cut in the middle of the character context.

LSTM achieved great success in many sequence labeling tasks. LSTM are the same as RNNs, except that the hidden layer updates are replaced by purpose-built memory cells. As a result, they may be better at finding and exploiting long range dependencies in the data. To get better effect, we employed bi-directional LSTM model (Graves et al. [8]) to get information of both sides of a word. The bi-LSTM architecture is shown in Figure 5.

For every candidate character sequence generated by new Apriori-like algorithm, firstly we get its character context in the corpus. For example, we assume that the window size is 4. As for character sequence $C_{\mathrm{t}}C_{\mathrm{t}+1}$, we find its per-position in the source text and acquire its two adjacent characters on the left and on the right respectively. So we can get characters window $C_{\mathrm{t}-2}C_{\mathrm{t}-1}C_{\mathrm{t}}C_{\mathrm{t}+1}C_{\mathrm{t}+2}$ $C_{\mathrm{t}+3}$. If there is no adjacent character, we replace it with symbol 'Padding'. Then, we can acquire three input context $C_{\mathrm{t}-2}C_{\mathrm{t}-1}C_{\mathrm{t}}C_{\mathrm{t}+1}$, $C_{\mathrm{t}-1}C_{\mathrm{t}}C_{\mathrm{t}+1}C_{\mathrm{t}+2}$, $C_{\mathrm{t}}C_{\mathrm{t}+1}C_{\mathrm{t}+2}C_{\mathrm{t}+3}$ respectively. Finally we can get the segmentation probabilities of these three contexts by LSTM. Thus we can get three segmentation probabilities of candidate character sequence $C_{\mathrm{t}}C_{\mathrm{t}+1}$ in per-position of the source corpus. The segmentation probability denotes the internal segmentation probability or boundary segmentation probability of a candidate word.

6.3 Training

We implement the neural network using the Keras framework. Character embeddings are trained by off-the-shelf tools word2vec. Training and inference are done based on segmented experiment corpus. The initial states of the LSTM are zero vectors.

Given a training set D, the regularized objective function is the loss function $J(\theta)$ including a ℓ_2-norm term:

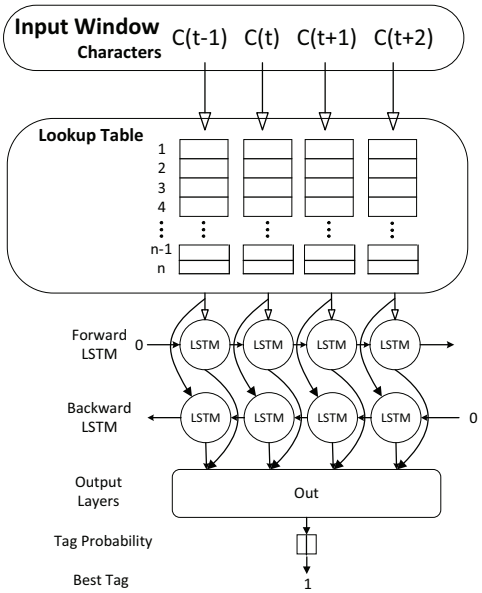


Fig. 5: The architecture of bi-LSTM model for segmentation probability

$$\ell_i(\theta) = \begin{cases} 0 & (y_i = Y_i) \\ 1 & (y_i \neq Y_i) \end{cases} \tag{1}$$

$$J(\theta) = \frac{1}{|D|} \sum_{(x_i, y_i) \in D} \ell_i(\theta) + \frac{\gamma}{2} \parallel \theta \parallel_2^2 \tag{2}$$

Where x_i, y_i are input and output of train set, Y_i denotes tag computed by the network, θ is the parameter set of our model, γ is a regularization parameter. We train our network to minimize the loss function using a generalization of gradient descent called subgradient method.

7 New Word Detection

In this section, we proposed a filtering rule and a statistical feature to filter out the noise new words $\mathcal{T}$ of candidate new words $\mathcal{A}$.

7.1 Filtering Rule

For every candidate character sequence generated by Apriori-like algorithm, Firstly we filter it based on a standard dictionary, if the standard dictionary contains it, we filtered out it and regarded it as a known word. Then, we would acquire candidate new words.

For every candidate new word, it may occur one or more times in the testing set. And there will be an input context in per-position. For every input context, it will have a segmentation probability sequence computed by LSTM network.

In the ancient Chinese literature, there are a lot of characters can be a single word. For this reason, we filter candidate character sequence based on below rule:

Rule 2: *For per-input context of character sequence, if there is one of context that both of its left and right adjacency segmentation probability are more than 0.5, means that it is segmented, we classify it to new word.*

7.2 Word Confidence

In order to analyze filtered new words better. We propose a word confidence (WC) feature, a probability score to measure how likely the new word is a valid word. It consists of two parts: Branching Probability (BP) and Cohesiveness Probability (CP).

$$BP(s) = \sqrt[2]{\sqrt[n]{\prod_{i=1}^{n} P_l i} * \sqrt[n]{\prod_{j=1}^{n} P_r j}} \tag{3}$$

$$CP(s) = \sqrt[n+l]{\prod_{i=1}^{n} \prod_{c=1}^{l-1} P_c} \tag{4}$$

$$WC(s) = \mu * BP(s) + (1 - \mu) * (1 - CP(s)) \tag{5}$$

Where n is the number of context of new word s, for each context, s has a segmentation probability sequence, P_l is its left adjacency segmentation probability and P_r is its right adjacency segmentation probability, P_c denotes internal segmentation probability. l is the length of new word s.

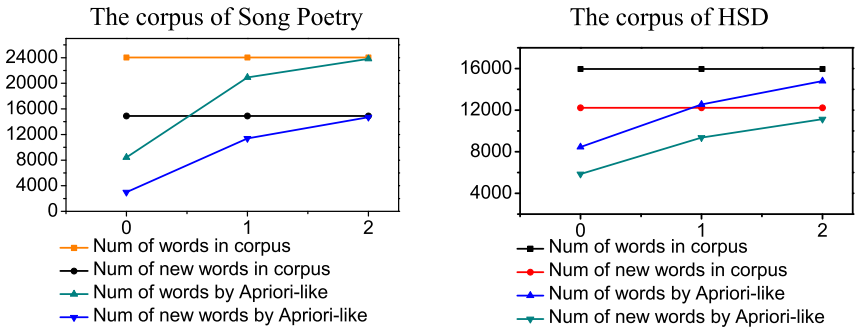
We add μ to the calculation of score because we find BP score is more important than CP score when defining whether new word s is a valid word.

8 Experiments

8.1 Datasets

In this paper, we mainly used two datasets, Song Poetry and History of the Song Dynasty, the most representative ancient Chinese literature. We crawled 19387 Song Poetry, consisting of around 2 million Chinese characters, from http://www.gushici.org, the largest ancient poetry website in China. And we acquired the History of the Song Dynasty, abbreviated as HSD, from experiment of Deng et al. [7]. HSD contains 496 chapters and about 5.3 million Chinese characters. Then, we asked four annotators to segment 30,000 random selected sentences in Song Poetry and 32,000 random selected sentences in HSD. If there was a disagreement, discussions were required to make the final decision.

In addition, we regarded the dictionary of jieba (a popular open source project in Github.com, contains 584429 known words) as a standard lexicon $\mathcal{D}$ to filter segmentation result. Then we manually selected words with specific semantics and acquired 14891 new words in Song Poetry and 12132 new words in HSD.



(a) (b)

Fig. 6: X-axis denotes the threshold of low frequency. Y-axis is the number of words generated by new Apriori-like algorithm.

8.2 Candidate New Words Generated by Apriori-like Algorithm

First, new Apriori-like algorithm was used to generate candidate character sequences in Song Poetry corpus and HSD corpus. In this experiment, the support were all set to 5. We set length of frequent itemsets to 5 for Song Poetry and set length of frequent itemsets to 10 for HSD, and the thresholds of low frequency were all set to 2. 99782 and 72255 candidate character sequences were generated in Song Poetry and HSD respectively. Then we analyzed these character sequences. Result is shown in Figure 6.

Number of words generated by Aprioir-like denotes how many character sequences belong to words of corpus. From Figure 6, we can find that improved Apriori-like algorithm can mine low frequency new words efficiently. And the recall rate of valid candidate character sequences has improved greatly.

8.3 Segmentation Probability by LSTM

We splitted 80% of tagged sequences as training set, left 20% as test set. Meanwhile, we set input context size to 4 and empirically set dropout rate to 0.26. Dropout is used to avoid over-fitting problem. We tested performance based on different character embedding dimensions and sizes of LSTM units in Song Poetry corpus and HSD corpus. Result is shown in table 3.

From table 3, we found that LSTM performed the best when character embeddings dimension is 200 and size of LSTM units is 150 in the corpus of HSD. In the corpus of Song Poetry, LSTM had the best performance when character embedding dimension is 128 and size of LSTM units is 100. Thus, we used this two sets of parameters to conduct the later experiments.

8.4 Detect All New Words in Corpus

We classified candidate character sequences generated by Apriori-like algorithm to training set and test set corresponding to LSTM network. In the test set, we

Table 3: Performance of LSTM network

Song Poetry Corpus					HSD Corpus				
Hiddens	Emb	P	R	F	Hiddens	Emb	P	R	F
150	50	90.72	90.61	90.66	150	50	91.80	91.61	91.66
128	100	90.96	90.86	**90.91**	128	100	91.67	91.50	91.55
150	100	90.71	90.66	90.68	150	100	92.12	92.01	92.05
200	100	90.83	90.76	90.79	200	100	91.90	91.73	91.77
150	150	90.30	90.22	90.06	150	150	92.06	91.91	91.96
150	200	90.68	90.52	90.60	150	200	92.19	92.09	**92.12**

ran improved Apriori-like algorithm and used the parameters of the previous experiments. Then we got 13905 candidate character sequences in Song Poetry and 12265 candidate character sequences in HSD. Then, the dictionary of jieba was used to filter candidate character sequences, we got 10173 candidate new words in Song Poetry and 4017 candidate new words in HSD.

We utilized LSTM network to compute segmentation probability of per-context of candidate new word. Then we filtered the result generated by LSTM network based on filtering rule. Result is shown in table 4.

Table 4: Result of new word detection

Corpus	Song Poetry	HSD
New word in corpus	2349	1778
Valid new word by our method	2107	1559
Detecting new word by our method	2852	2193
Precision	89.70	87.68
Recall	73.88	71.74
F1 value	81.02	78.92

As we can learn from the table 4, our recall rate is relatively low. It mainly caused by the accuracy of LSTM network. Wrong tags will misled the filtering result. In addition, we analyzed the invalid new words and found that some new words did not occur in the test set but occur in the training set and un-annotated corpus, which further illustrates our method is effect and precise.

8.5 Word Confidence Analysis

We further analyzed the results of new word detection in two corpus. Word confidence score of each candidate new word was computed and ranked in Figure 7.

It can be found that our word confidence feature is effect in word recognition. When we obtained top 2000 words of ranking candidate new words in the two corpus, the accuracy could be up to 90%. And we listed the top 20 new words ranked by word confidence score. It is shown in table 5.

As is shown in table 5, the quality of ranking result was very high. Compound words (e.g., '槛菊', it means 'Chrysanthemum outside the railing') and low frequency words (e.g., '亓氏', it is a name) could be discovered well.

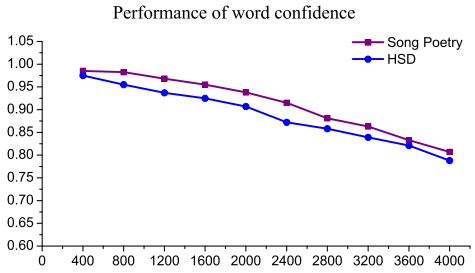


Fig. 7: Result of word confidence score ranking. X-axis is the number of words returned (K), and Y-axis is precision of valid new word.

Table 5: Top 20 new words in corpus

Song Poetry										HSD									
1	*流霞*	5	*沙汀*	9	*嫩香*	13	*孤帏*	17	*万斛*	1	*亓氏*	5	*是歲*	9	*交州*	13	*升黜*	17	*岢嵐*
2	*寒灯*	6	*宝辇*	10	*苹花*	14	*岁华*	18	*井梧*	2	*孫覺*	6	*畿内*	10	*暴骸*	14	*臺諫*	18	*徹樂*
3	*洞户*	7	*残蝉*	11	*红茵*	15	*画檐*	19	*幽闺*	3	*浮漏*	7	*龜茲*	11	*賢妃*	15	*熟戶*	19	*伏誅*
4	*槛菊*	8	*万灵*	12	*九仪*	16	*庭宇*	20	*星闱*	4	*銀帛*	8	*冢宰*	12	*郊宮*	16	*夏兵*	20	*熒惑*

8.6 AP-LSTM vs. Other Technique

We compared AP-LSTM with the current state-of-the-art open source Chinese segmentation tools (Ansj, ICTCLAS, and Stanford Chinese-word-segmenter) in Song Poetry corpus. And we also compared our result with TopWords model (Deng et al. [7], they showed that the accuracy of their model has reached 90%) in HSD corpus.

We used segmentation tools to segment corpus and filtered segmentation result with the dictionary of jieba. Then we counted the number of new words found by segmenter. In fairness to all tools, we transform the ancient Chinese to simplified Chinese by Opencc[1] in the first group of experiments. Comparison results is show in Table 6.

Table 6: Comparative results of AP-LSTM with other Technique on new word detection

Corpus	Song Poetry				HSD	
Category	AP-LSTM	Ansj	ICTCLAS	Chinese-word-segmenter	AP-LSTM	TopWords
Noise words	242	2219	2219	1551	219	372
Valid new words	**2107**	130	130	798	**1559**	1406
Low frequency words	**243**	50	45	77	**207**	112
Precision	**89.70**%	5.53%	5.53%	33.97%	**87.68**%	79.08%

As shown in the table 6, the performance of AP-LSTM is much better than other open source Chinese segmentation tools in Song Poetry corpus, which illustrated there were great difference between modern Chinese and ancient Chinese.

[1] https://github.com/BYVoid/OpenCC

The tools and model of modern Chinese may not be suitable for ancient Chinese. In the corpus of HSD, AP-LSTM performs better than TopWords, we compared the result of discovering words and found that our method could find more low frequency words and reduce segmentation ambiguity. In the result of TopWords, there was segmentation ambiguity in many words, which results in the various segmentation forms of a same word and the severe dependence of the accuracy of TopWords on sample data.

8.7 Experiment on Tokenizer

As we have detect new words in the corpus, we add all these new found words in the dictionary of tokenizer, and compare its performance with that of original tokenizer. The experiment is operated on the segmented subset of corpus and its result is as follow:

Table 7: Segmentation Evaluation

Segmentation Evaluation	Evaluation		
	Precision	*Recall*	$F1$
Original tokenizer	0.6544	0.6149	0.6340
Add new words	0.8213	0.8025	0.8117

From the table above, the adding of unknown words greatly improves the performance of tokenizer.

9 Conclusion

In this paper, we proposed a new word detection model based on improved Apriori-like algorithm and LSTM network in ancient Chinese literature. The AP-LSTM model can find low frequency new words efficiently. And word confidence feature can further mine new words which makes result more accurate. Finally, we detected new words in two of the most representative ancient Chinese literature, Song Poetry and History of the Song Dynasty. Experiments show the effectiveness of AP-LSTM model.

In the future work, we will add more character features to character embeddings, which can improve the performance and accuracy of neural network.

Acknowledgement

This work is supported in part by the National Basic Research(973) Program of China (No.2013CB329606). The authors would like to thank Xinyu Wu, Chunzi Wu, Chang Liu and Zhao Tang for their help in tagging, and Bin Wu for his advice to this paper.

References

1. Agrawal, R., Srikant, R., et al.: Fast algorithms for mining association rules. In: Proc. 20th int. conf. very large data bases, VLDB. vol. 1215, pp. 487–499 (1994)

2. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. journal of machine learning research 3(Feb), 1137–1155 (2003)
3. Bu, F., Zhu, X., Li, M.: Measuring the non-compositionality of multiword expressions. In: Proceedings of the 23rd International Conference on Computational Linguistics. pp. 116–124. Association for Computational Linguistics (2010)
4. Chen, A.: Chinese word segmentation using minimal linguistic knowledge. In: Proceedings of the second SIGHAN workshop on Chinese language processing-Volume 17. pp. 148–151. Association for Computational Linguistics (2003)
5. Chen, K.J., Ma, W.Y.: Unknown word extraction for chinese documents. In: Proceedings of the 19th international conference on Computational linguistics-Volume 1. pp. 1–7. Association for Computational Linguistics (2002)
6. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. Computational linguistics 16(1), 22–29 (1990)
7. Deng, K., Bol, P.K., Li, K.J., Liu, J.S.: On the unsupervised analysis of domain-specific chinese texts. Proceedings of the National Academy of Sciences p. 201516510 (2016)
8. Graves, A., Fernández, S., Schmidhuber, J.: Bidirectional lstm networks for improved phoneme classification and recognition. In: International Conference on Artificial Neural Networks. pp. 799–804. Springer (2005)
9. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Networks 18(5), 602–610 (2005)
10. Huang, M., Ye, B., Wang, Y., Chen, H., Cheng, J., Zhu, X.: New word detection for sentiment analysis. In: ACL (1). pp. 531–541 (2014)
11. Justeson, J.S., Katz, S.M.: Technical terminology: some linguistic properties and an algorithm for identification in text. Natural language engineering 1(01), 9–27 (1995)
12. Peng, F., Feng, F., McCallum, A.: Chinese segmentation and new word detection using conditional random fields. In: Proceedings of the 20th international conference on Computational Linguistics. p. 562. Association for Computational Linguistics (2004)
13. Sun, X., Wang, H., Li, W.: 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: Long Papers-Volume 1. pp. 253–262. Association for Computational Linguistics (2012)
14. WANG, L.x., WANG, J.d., WANG, J.: Approach for lexicon updating based on data mining [j]. Application Research of Computers 12, 062 (2006)
15. Zhang, H., Shi, S.: Which performs better for new word detection, character based or chinese word segmentation based? In: Asian Language Processing (IALP), 2014 International Conference on. pp. 10–14. IEEE (2014)
16. Zhang, W., Yoshida, T., Tang, X., Ho, T.B.: Improving effectiveness of mutual information for substantival multiword expression extraction. Expert Systems with Applications 36(8), 10919–10930 (2009)
17. Zhang, Y., Sun, M., Zhang, Y.: Chinese new word detection from query logs. In: International Conference on Advanced Data Mining and Applications. pp. 233–243. Springer (2010)
18. Zheng, Y., Liu, Z., Sun, M., Ru, L., Zhang, Y.: Incorporating user behaviors in new word detection. In: IJCAI. vol. 9, pp. 2101–2106. Citeseer (2009)