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. Institute 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, especially in Tang’s epitaph texts (achieving an F1-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...
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.

![A classical Chinese texts example: it is a famous prose from dynasty Song](image)

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 with part-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.

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 ‘’ (moon) ‘mutated’ from the character for meat) usually means body parts as a radical such as ‘’ (leg): u8170 and ‘’ (arm): u818a. Some simplified Chinese characters lost some radicals. For example the upper radical of the character ‘’ (cloud: u96f2) is ‘’ which means rain; whereas the simplified one ‘’ 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.

\[
L = \sum_{i} \log P(x_i|h_i)
\]

where \(x_i\) is the output Chinese character \(h_i\) is the concatenation of \(c_i\) and \(r_i\).

\[
h_i = \text{cat}(c_{i-1} \ldots c_{i+w} r_{i-1} \ldots r_{i+w})
\]

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 \(W\). Different from the original CBOW model the extra parameter introduced in the matrix \(W\) 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 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_{i-1} and c_{i-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.

\[ i_t = \sigma(W_i x_t + W_{ih} h_{t-1} + W_{ic} c_{t-1} + b_i) \]  
\[ f_t = \sigma(W_f x_t + W_{fh} h_{t-1} + W_{fc} c_{t-1} + b_f) \]  
\[ c_t = f_t c_{t-1} + i_t \tanh(W_x x_t + W_{hc} h_{t-1} + b_c) \]  
\[ o_t = \sigma(W_o x_t + W_{ho} h_{t-1} + W_{oc} c_t + b_o) \]  
\[ h_t = o_t \tanh(c_t) \]

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. \( \sigma \) 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.

\[ s(x[y]) = \sum_{i=1}^{n} A_{i,j} y_i + \sum_{i=1}^{n} p_i y_i \]  

The input matrix \( P \) is of size \( n \times k \) where \( k \) is the number of tagging. \( A \) 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 \( x = x_1 x_2 \ldots x_n \) and the output is a vector \( y = y_1 y_2 \ldots y_n \) where each \( y_i \) is a probability vector corresponding to each tagging. Fig. 4 shows the detail of our algorithm.

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>
<thead>
<tr>
<th>Dynasty</th>
<th>Total characters</th>
<th>Vocabulary size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tang</td>
<td>6 160 233</td>
<td>7 478</td>
</tr>
<tr>
<td>Ming</td>
<td>7 414 125</td>
<td>9 952</td>
</tr>
<tr>
<td>Qing</td>
<td>29 392 770</td>
<td>1 0151</td>
</tr>
<tr>
<td>Tang-epitaph</td>
<td>1 116 850</td>
<td>5 197</td>
</tr>
</tbody>
</table>

We divide out texts according to dynasties and literary styles: dynasty Tang (A.D. 618 — 907) 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 uses begin-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.

![Fig. 5 Tagging example](image)

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 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>
<thead>
<tr>
<th>Hyper parameters</th>
<th>Word embedding dimension</th>
<th>Hidden layer size</th>
<th>Hidden layer</th>
<th>Batch</th>
<th>Epoch</th>
<th>Learning rate</th>
<th>Gradient clipping</th>
<th>Dropout rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word embedding dimension</td>
<td>100</td>
<td>100</td>
<td>1</td>
<td>50</td>
<td>30</td>
<td>0.01</td>
<td>5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

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 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.

$$P = \frac{TP}{TP + FP}$$

(9)

$$R = \frac{TP}{TP + FN}$$

(10)
$F_1 = \frac{2PR}{P + R}$  \hspace{1cm} (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 addition 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 better training and a better definition of sentence structures (according to the modern Chinese rules) of the texts.

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Sentence segmentation results for Tang</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tang</td>
<td>$P$</td>
</tr>
<tr>
<td>CRF</td>
<td>0.757</td>
</tr>
<tr>
<td>LSTM</td>
<td>0.766</td>
</tr>
<tr>
<td>Bi-LSTM-CRF</td>
<td>0.734</td>
</tr>
<tr>
<td>Bi-LSTM-CRF with radical embedding</td>
<td>0.747</td>
</tr>
</tbody>
</table>

Table 4 | Sentence segmentation results for Ming |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ming</td>
<td>$P$</td>
</tr>
<tr>
<td>CRF</td>
<td>0.624</td>
</tr>
<tr>
<td>LSTM</td>
<td>0.669</td>
</tr>
<tr>
<td>Bi-LSTM-CRF</td>
<td>0.722</td>
</tr>
<tr>
<td>Bi-LSTM-CRF with radical embedding</td>
<td>0.696</td>
</tr>
</tbody>
</table>

Table 5 | Sentence segmentation results for Qing |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Qing</td>
<td>$P$</td>
</tr>
<tr>
<td>CRF</td>
<td>0.779</td>
</tr>
<tr>
<td>LSTM</td>
<td>0.686</td>
</tr>
<tr>
<td>Bi-LSTM-CRF</td>
<td>0.741</td>
</tr>
<tr>
<td>Bi-LSTM-CRF with radical embedding</td>
<td>0.761</td>
</tr>
</tbody>
</table>

Table 6 | Sentence segmentation results for Tang epitaph |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tang-epitaph</td>
<td>$P$</td>
</tr>
<tr>
<td>CRF</td>
<td>0.803</td>
</tr>
<tr>
<td>LSTM</td>
<td>0.759</td>
</tr>
<tr>
<td>Bi-LSTM-CRF</td>
<td>0.789</td>
</tr>
<tr>
<td>Bi-LSTM-CRF with radical embedding</td>
<td>0.814</td>
</tr>
</tbody>
</table>

Taking the epitaph of the dynasty 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’.

![Graph showing 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.

<table>
<thead>
<tr>
<th>Our model:</th>
<th>Prof. Su is 47 years old</th>
<th>in Province Huang</th>
<th>lives in pavilion Linao</th>
<th>builds a house near Dongpo (Location)</th>
<th>called Dongpo hermit by himself</th>
<th>based on the map of Dongpo</th>
<th>from the south door of Province Huang</th>
<th>430 feet to Xuetang</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orginal:</td>
<td>Prof. Su is 47 years old</td>
<td>in Province Huang</td>
<td>lives in pavilion Linao</td>
<td>builds a house near Dongpo (Location)</td>
<td>called Dongpo hermit by himself</td>
<td>based on the map of Dongpo</td>
<td>from the south door of Province Huang</td>
<td>Xuetang is 430 feet from the south door of province Huang</td>
</tr>
</tbody>
</table>

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


(Editor: Ai Lisha)