A Machine Learning Model for the Dating of Ancient Chinese Texts

Xuejin Yu
University of Science & Technology Beijing
Beijing, China
yuxuejin94@gmail.com

Wei Huangfu
University of Science & Technology Beijing
Beijing, China
huangfuwei@ustb.edu.cn

***Abstract*—This paper, with the intent of solving the issues on the dating of ancient Chinese texts, takes advantage of the Long-Short Term Memory Network (LSTM) to analyze and process the character sequence in ancient Chinese. In this model, each character is transformed into a high-dimensional vector, and then vectors and the non-linear relationships among them are read and analyzed by LSTM, which finally achieve the dating tags. Experimental results show that the LSTM has a strong ability to date the ancient texts, and the precision reaches about 95% in our experiments. Thus, the proposed model offers an effective method on how to date the ancient Chinese texts. It also inspires us to actively improve the time-consuming analysis tasks in the Chinese NLP field.**

***Keywords*-Ancient Chinese texts; Dating; Machine learning; LSTM**

I. Introduction

Natural Language Processing (NLP) is a cross-disciplinary research field of linguistics, computer science, information engineering, and artificial intelligence. With such technologies, we process and analyze large amounts of natural language data by means of computers. In recent years, Chinese natural language processing, as an important part of NLP, has increasingly attracted attention of academic world, and made gratifying achievements. Ancient Chinese texts, carrying philosophy, culture, knowledge, wisdom and spirit of the Chinese people during thousands of years, are usually referred to as the Pearl of the crown of Chinese language. However, there are many essential issues to be solved, ranging from digitization, labeling, to categorization for such ancient Chinese texts. Thus we need to use advanced Chinese natural language processing technologies to further process and perform data mining. It will be beneficial to the preservation and knowledge extraction of existing ancient Chinese texts, and greatly promotes the construction of digital humanities in China.

The current progress of digitization of ancient Chinese texts mostly remains in the early stage. There are short of studies on part-of-speech tagging, named entity recognition, text structure processing, text classification, etc. Moreover, the accuracy of the existing methods on part-of-speech tagging is not enough since the Chinese ancient texts can be traced back to the Shang dynasty (at least more than 3,000 years ago). In the evolution of these years, the meaning, the grammar and the syntax of the Chinese language are constantly changing.

Take "汤" (tang in pinyin), an ancient Chinese character, as an example. It originally means "hot water", but now only refers to the soup obtained after cooking the food. Therefore, even a character in different periods possibly has different meanings.

The Chinese language has a long history, which leads to great difficulties in constructing a suitable model for various periods. Therefore, only by judging the approximate time of the ancient texts, can we conduct subsequent studies on ancient texts and further improve the accuracy and efficiency of the language research.

In this paper, we attempt to explore the field of NLP for the ancient Chinese texts from the perspective of the text dating. We focus on the problem of the dating of ancient Chinese texts with deep learning network model. With our approach, there is no need to manually extract rule features. The research results of this paper will help to study the ancient Chinese word segmentation, part-of-speech tagging, text structure processing, text classification and so forth.

II. Related Work

From a technical point of view, the time judgment of ancient text is a typical text classification task.

Currently, text classification methods can be roughly categorized into two categories: the former is the traditional machine learning methods based on rules or probability, and the latter is the deep learning methods based on Convolutional Neural Networks (CNN) [1], Recurrent Neural Networks (RNN) [2] or self-Attention [3].

The rule-based and the probability-based approaches are relatively simple, easy to implement, and work well in many specific areas. However, in the rule-based and the probability-based methods, many rules or specific conditions need to be considered, so it is necessary to define and manually extract features by experts.

Recently, deep learning algorithms are widely used to solve the language processing problems. Collobert [4] and Tang [5] apply CNN and RNN to process natural language, respectively. The Bidirectional Encoder Representations from Transformers (BERT) model is proposed based on self-attention mechanism in 2018. It performs very well in lots of kinds of typical NLP tasks. However, the BERT model is mainly designed for modern languages, and depends on the massive informational texts of the current Internet era, such as Wiki encyclopedia entries, various news media, and commentary messages. However, it is not feasible in the ancient Chinese research field where the corpus resources are relatively limited.

III. Ancient Chinese Text Dating Model

We try to model the problem of time determination of ancient Chinese texts as follows:

$$T = M\Big(\{v_1, v_2, \ldots, v_n\}\Big) = M\Big[g\Big(\{x_1, x_2, \ldots, x_n\}\Big)\Big] \tag{1}$$

where $g(\cdot)$ is a mapping function, $\{x_1, x_2, \ldots, x_n\}$ is a piece of Chinese texts to be determined, and $\{v_1, v_2, \ldots, v_n\}$ denotes its vector sequence [6]. Then the vector sequence is processed by the model M, which calculates the outputs label T.

In this paper, we use Google's word2vec [7] model to obtain word vectors. Assumed that if the contexts of two words are similar, then their semantics are similar in such a model. The word2vec, an unsupervised model, is able to obtain the result in a relatively short period at lower cost. We can learn vector representation of the word in a large number of unlabeled corpora through the continuous bag of words (CBOW) model of word2vec. The input of the CBOW model is the word vectors of words in the context of the central word, and the output is the word vector of the given word. The output vector contains the relationships between words. After getting the output vector with the relationship, we use it as a feature for next steps to improve the generalization ability of the model.

The neural network that is commonly seen is like a spider web that converges from many nodes to a single output. Here we have a single input and a single output. Such a network works well for non-continuous inputs, where the order of the inputs does not affect the output.

In the text processing, the order of the characters is very important. The RNN can accept continuous input, using the activation of the previous node as the parameter of the latter node [8]. However, RNNs are not very good at passing information from very early units. The Long-Short Term Memory Network (LSTM) uses a memory unit to store certain information that occurred before [9]. A memory unit in LSTM includes input gates, output gates, and forgetting gates to control the preservation of information, which makes LSTM better in the language model.

A. Network Structure

The block diagram of the proposed ancient Chinese dating model is shown in Fig. 1.

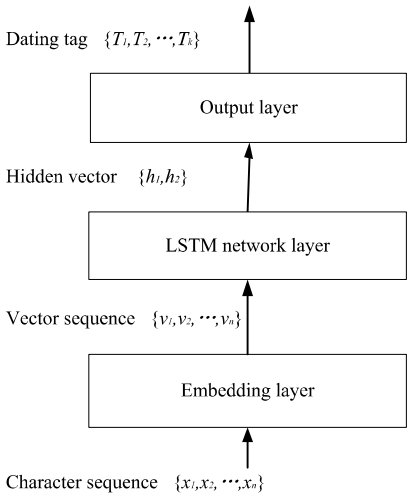

Figure 1. The block diagram of the ancient Chinese dating model.

As shown in Fig. 1, the model accept a piece of text as its input and predict its occurrence time as its output. First, a piece of text to be predicted is obtained, and the text is sent to the embedding layer to obtain a vector representation of the text. Then the character vectors are sent to the LSTM neural network layer to calculate the hidden vector. Finally, a fully connected network calculates the hidden vector and outputs the time stamp labels.

The description of the model in detail is shown in Fig. 2. The model consists of 3 layers, which are the embedding layer, the LSTM forward and backward layer(s), and the output layer. The embedding layer is a CBOW model of word2vec, which represents Chinese characters as a sequence of vectors. Then there are the bidirectional-LSTM [10], [11] layers, which are the main body of the model. Their forward and reverse inputs are the vector sequences in different orders, respectively. Both the forward and reverse parts of the LSTM network output part of the hidden vectors. The output layer concatenates the two hidden vectors and sends them to a fully connected layer. Finally, the different dating tags are determined by the probability with a Softmax operation. It is noting that the hidden vector is changed while the word vector is fed to LSTM since the LSTM networks are for time series.

B. LSTM Memory Unit

LSTM exploits long-term dependencies by using input gates, input gates, and forgetting gates. The typical schematic of LSTM memory unit is shown as Fig. 3

Fig. 3 shows the LSTM neural network expanding in time dimension. The formulae of the LSTM memory unit are as follows:

$$f^{(t)} = \sigma\Big(W_{fv}v^{(t)} + W_{fh}h^{(t-1)} + W_{fc}c^{(t-1)}\Big) \tag{2}$$

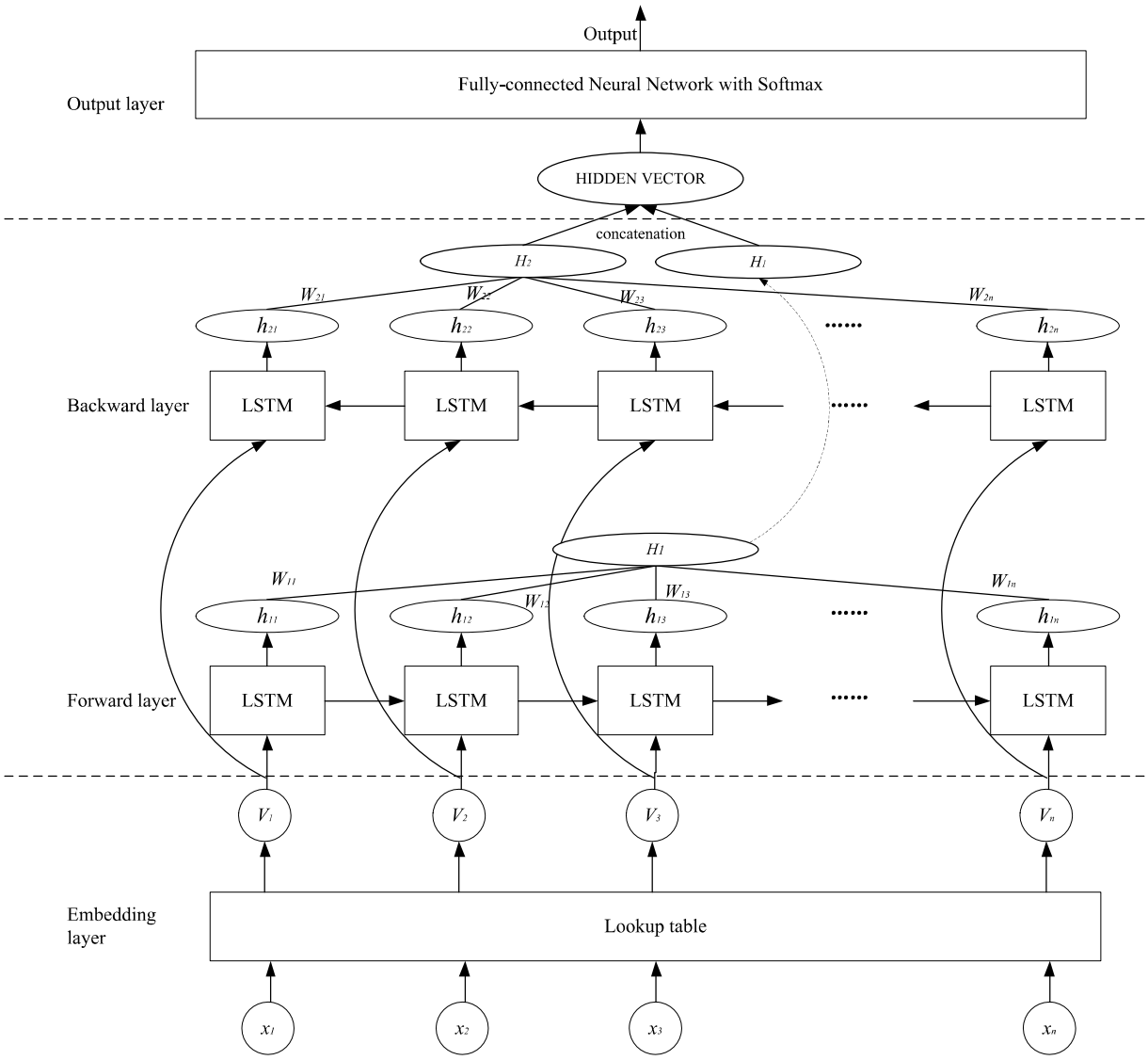

Figure 2. The Bi-LSTM structure diagram.

$$i^{(t)} = \sigma\left(W_{iv}v^{(t)} + W_{ih}h^{(t-1)} + W_{ic}c^{(t-1)}\right) \quad (3)$$

$$a^{(t)} = \Phi\left(W_{cv}v^{(t)} + W_{ch}h^{(t-1)}\right) \quad (4)$$

$$c^{(t)} = f^{(t)} \cdot c^{(t-1)} + a^{t} \cdot i^{(t)} \quad (5)$$

$$o^{(t)} = \sigma\left(W_{ov}v^{(t)} + W_{oh}h^{(t-1)} + W_{oc}c^{(t)}\right) \quad (6)$$

$$h^{(t)} = o^{(t)} \cdot \Phi(c^{(t)}) \quad (7)$$

In these formulae, $i^{(t)}$ is the input gate, σ is the sigmoid function, and the function of the sigmoid function is to make each element of the output vector valued in the interval $[0, 1]$. Also, Φ is a tanh function, which is used to make each element of the output vector between $[1, -1]$; $f^{(t)}$ is the forgetting gate; $c^{(t)}$ is used to store the long-term information, which is obtained by adding the product of the information; $c^{(t-1)}$ is for the last moment; the forgetting gate $f^{(t)}$ is used to the product of the current input state $a^{(t)}$ and the input gate $i^{(t)}$; $o^{(t)}$ is the output gate to control output vectors; $h^{(t)}$ is the output of the current moment, which is obtained by multiplying the output gate $c^{(t)}$ by the current state of the information $\Phi(c^{(t)})$.

C. Output Layer

The output layer is a fully connected network. It accepts a vector h of $1 \times 2n$ dimensions, which is generated by the LSTM layer, while n is the dimensions of hidden vector. The output layer has a weight w vector of $2n \times k$

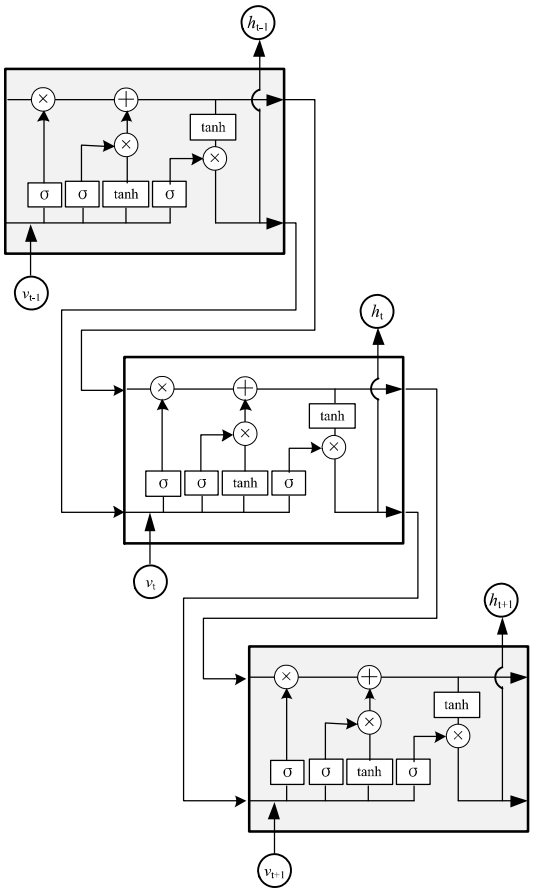

Figure 3. The LSTM memory unit structure.

dimensions and an offset b vector of $1 \times k$ dimensions, where k is the number of the total classification tags. A $1 \times k$ dimensional vector output is obtained by the formula $y = Wx + b$. The value of each of these dimensions can be seen as the likelihood of the kind represented by that dimension.

The Softmax function is used to normalize the vector elements of the output layer output, and the normalized probability is output. Here the Softmax function is given by

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_k e^{z_k}}. \tag{8}$$

IV. EXPERIMENTS AND DISCUSSION

The corpus used in the experiments is some ancient texts of the Spring and Autumn period and the Warring States period. We divided them into three small time periods: the Spring and Autumn Period, Early Warring States Period and Late Warring States Period, as shown in Table I. We use the symbols T_1, T_2, T_3, to represent the three periods, respectively.

In each time period, we selected some ancient texts as the source of the training data. Totally 8 ancient works

Table I
DEFINITION OF LABELS

Tag	Year	Dynasty
T1	Before 475 B.C.	the Spring and Autumn Period
T2	475 B.C.–350 B.C.	Early Warring States Period
T3	350 B.C.–221 B.C.	Late Warring States Period

Table II
THE SOURCE OF THE TRAINING DATA.

Tag	Book Name	Number of characters (×1000)
T1	*Shangshu*	45
T1	*Chunqiu*	32
T1	*Yili*	107
T1	*Zhouli*	90
T2	*Zhouyi*	5
T2	*Mu Tianzi Biograph*	17
T3	*Zuozhuan*	381
T3	*Guoyu*	130

are loaded as shown in the following Table II. It covers totally about 800,000 Chinese characters.

Our training procedure is shown in Fig. 4.

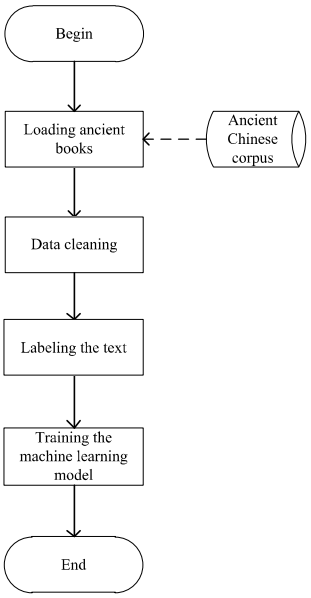

Figure 4. The schematic diagram of the training process.

A. Parameter selection

We conduct experiments to compare different hyper parameters. Different combinations of the parameters for the Hidden Layer Dimension (HLD) and the word Embedding Layer Dimension (ELD) are considered. We compare the accuracy of the training process under different parameter combinations. Fig. 5 shows the classification accuracy curves of the training process. It means that the convergence is the fastest when the HLD is 64 and the ELD is 64.

We also compare the accuracy indexes of the proposed model on the test data under different parameter combi-

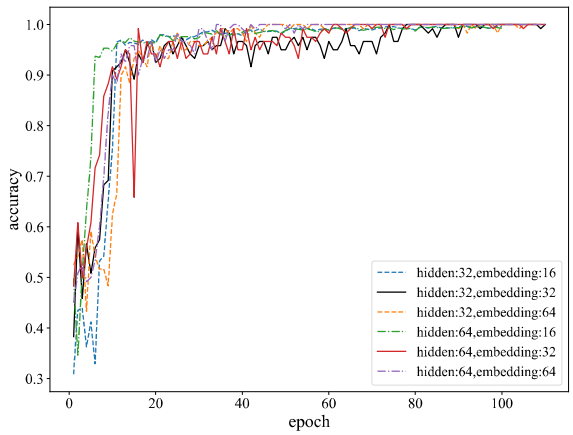

Figure 5. The classification accuracy curve of the training process.

nations. The experimental results are shown in Table III. Note that the following experiments are assigned to a hidden layer dimension of 32 and a word embedding vector dimension of 32.

Table III
THE ACCURACY ON THE TEST DATA.

HLD	ELD	Accuracy
32	16	0.971
32	32	0.987
32	64	0.960
64	16	0.975
64	32	0.962
64	64	0.975

B. Experiments

In the first experiment, we select some chapters which are not fed into the training model for each book. After the training procedure, we feed those non-training sentences to the network to judge their tags. Each line in the table below indicates the number of sentences in a given period that are judged to various tags. Table IV shows the results. In this case, the probability that a sentence is judged to be the correct tags is large, which means the proposed model performs well in the text dating.

Table IV
THE RESULT OF EXPERIMENT ONE.

Input \ Output	T1	T2	T3	Total
T1	1148(76%)	142(10%)	214(14%)	1504(100%)
T2	123(9%)	1092(85%)	75(6%)	1290(100%)
T3	47(4%)	10(1%)	1089(95%)	1146(100%)

We also try to test a book not from the same source. We use the rest of the books labeled with the same tag as the training set to observe the classification effect of the model on the whole texts of the non-training book. Since the proposed model takes text paragraphs as its input, we divide a book into many paragraphs. Then we judge the dating results of these paragraphs.

We conducted the experiment with "*Zuo Zhuan*" as an example. The judgment result is as Fig. 6. It can be seen that 994 pieces of the texts in "*Zuo Zhuan*" are labelled T1, 529 are labelled the T2, and 2132 are labelled T3. On the whole, the proposed model judges that the books "*Zuo Zhuan*" should be labelled with T3. This is consistent with our knowledge in the ancient Chinese community and it also confirms the correctness of our model.

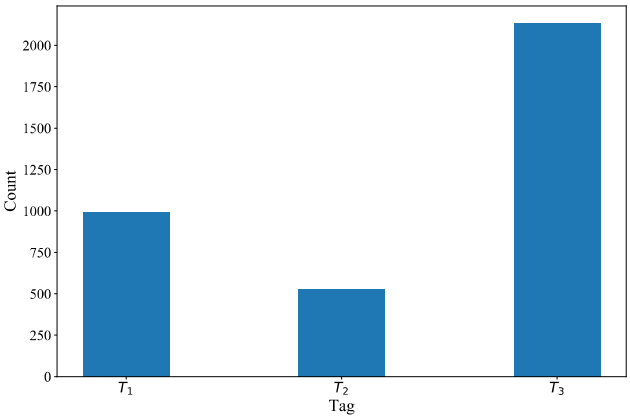

Figure 6. The bar chart of "*Zuo Zhuan*" dating results.

The experimental results show that if some texts of an ancient book are selected in the training set and the rest are used for the measurement set, the experimental correct rate of the proposed model is about 95%. However, if the ancient books are not involved in the training set, the correct rate of the paragraphs of the ancient books will be reduced. The proposed model also correctly labels the book if all the paragraphs in this book is fed into the model.

V. CONCLUSION AND FUTURE WORK

In this paper, we have utilized the LSTM to date the ancient Chinese texts. LSTM has great ability to analyze and recognize natural language, and we even show LSTM is outstanding in ancient Chinese processing. Our experiment results show that the precision of the LSTM reaches 95% in the dating of ancient Chinese books. Thus, the proposed model offers an effective method on how to date the ancient Chinese texts. It also inspires us to actively improve the time-consuming analysis tasks in the Chinese NLP field.

It is worth noting that the model we proposed is only a small exploration in the field of the NLP technologies related to ancient Chinese studies. There are still many shortcomings that need to be further improved in the future, including but not limited to, increasing the size of the data set, improving the word vector embedding and conducting more extensive experiments.

ACKNOWLEDGE

This research was funded by the Social Science Foundation Project of Beijing (Grant No. 18YYB003). The corresponding author is Dr. Wei Huangfu.

REFERENCES

[1] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[2] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[4] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of machine learning research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[5] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Pro. EMNLP*, Lisbon, Portugal, Sep. 2015, pp. 1422–1432.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[7] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Advances in neural information processing systems*, 2014, pp. 2177–2185.

[8] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. INTERSPEECH 2010*, Lyon, France, 2010.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[11] C. Dong, J. Zhang, C. Zong, M. Hattori, and H. Di, "Character-based lstm-crf with radical-level features for chinese named entity recognition," in *Natural Language Understanding and Intelligent Applications*. Springer, 2016, pp. 239–250.