Two Sequence Labeling Approaches to Sentence Segmentation and Punctuation Prediction for Classic Chinese Texts

Xuebin Wang and Zhenghua Li
School of Computer Science and Technology, Soochow University, China
xbwang15@stu.suda.edu.cn; zhli13@suda.edu.cn

Abstract

This paper describes our system for the EvaHan2024 shared task. We design and experiment with two sequence labeling approaches, i.e., one-stage and two-stage approaches. The one-stage approach directly predicts a label for each character, and the label may contain multiple punctuation marks. The two-stage approach divides punctuation marks into two classes, i.e., pause and non-pause, and separately handles them via two sequence labeling processes. The labels contain at most one punctuation marks. We use pre-trained SikuRoBERTa as a key component of the encoder and employ a conditional random field (CRF) layer on the top. According to the evaluation metrics adopted by the organizers, the two-stage approach is superior to the one-stage approach, and our system achieves the second place among all participant systems.

Keywords: EvaHan2024, Sentence Segmentation, Punctuation Prediction, Sequence Labeling

1. Introduction

One important characteristic of classic Chinese texts if the lack of punctuation marks. Readers have to decide sentence boundaries. In consequence, an article in classic Chinese is usually much more ambiguous than that in modern Chinese. The goal of the EvalHan2024 shared task is to see whether computation models can automatically perform sentence segmentation (SS) and punctuation prediction (PP).

We design and experiment with two sequence labeling approaches, i.e., one-stage and two-stage approaches. The one-stage approach is quite straightforward. It directly predicts a label for each character, and the label may contain multiple punctuation marks, as shown in the bottom row in Figure 2.

For the two-stage approach, we distinguish two types of punctuation marks, i.e., pause and non-pause, as shown in Table 1 . Then, we predict the two types of punctuation marks using two separate sequence labeling models. For both models, each label contains at most one punctuation mark.

Pause marks corresponds to those indicating sentence boundaries. Therefore, once the punctuation marks are obtained, we can infer sentence boundaries. Therefore, we only focus on the PP subtask, and solve the SS subtask as byproduct.

For the model architecture, we employ a standard conditional random field (CRF) model, using SikuRoBERTa as a key component of the encoder, as shown in Figure 1.

According to the evaluation metrics adopted by the organizers, the two-stage approach is superior to the one-stage approach, and our system achieves the second place among all participant systems. Compared to the baseline model Xunzi-Qianwen-7B-CHAT, our models obtain large improvement. Our code is available at https://github.com/XuebinWang-ai/EvaHan2024_PP.

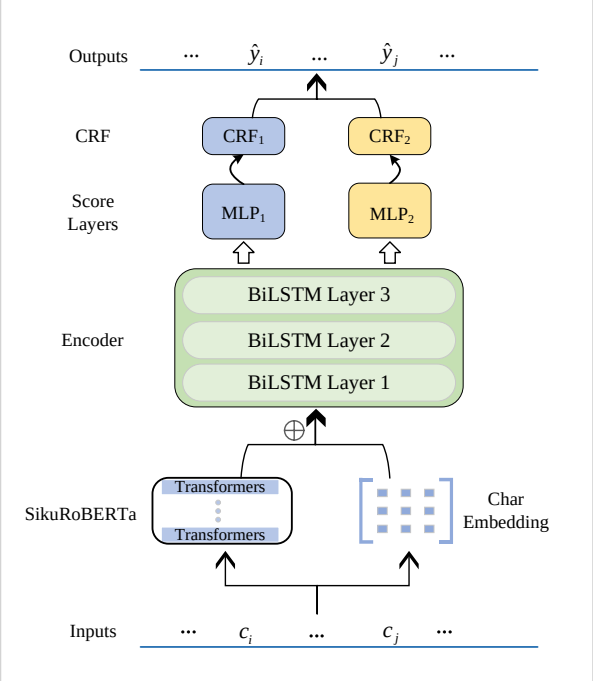

Figure 1: Model architecture.

2. Related Works

Sentence Segmentation & Punctuation Prediction A work by Xu et al. (2019) demonstrates combining word embedding and radical embedding can enhance the LSTM-CRF model in the SS task. A research by Hu et al. (2021) indicates a notable improvement in the performance of the BERT language model (Devlin et al., 2019) compared to the BiLSTM-CRF model in the SS task,

LT4HALA 2024@LREC-COLING-2024, pages 237–241

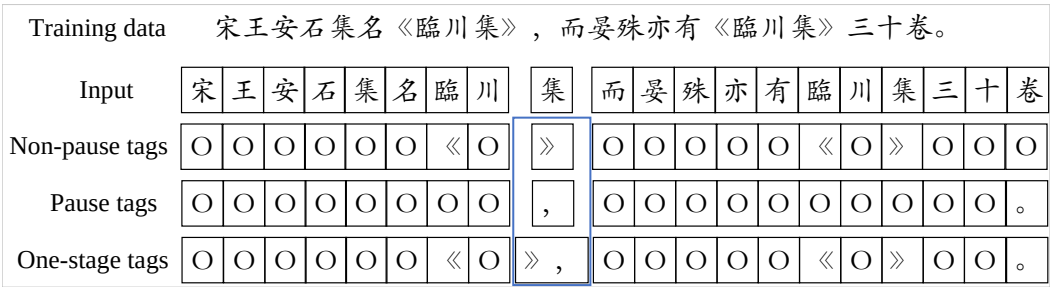

Figure 2: This excerpt is from the pre-processed training dataset. Punctuation marks are typically annotated on the characters to their left, apart from three specific types of left punctuation marks, which are annotated on the characters to their right. The "O" tags represent positions without punctuation.

Pause marks		Non-pause marks	
Name	Punc	Name	Punc
Comma	，	Double Quotation	“ ”
Period	。	Single Quotation	‘ ’
Slight-pasue	、	Book Title Marks	《 》
Question	？		
Exclamation	！		
Colon	：		
Semiclon	；		

Table 1: Pause and non-pause punctuation marks.

Symmetrical pairs			
Punc pair	Number	Punc pair	Number
。”	55580	”。	3293
？”	17878	”？	63
！”	8447	”！	32
。’	1945	’。	417
。》	843	》。	3043
，”	138	”，	6899
，》	35	》，	4957

Table 2: High frequency punctuation pairs.

resulting in a remarkable 10% increase in the F1 score. Conversely, a study by YuJ highlightes that the use of the BERT-BiLSTM-CRF model slightly improves the PP task performance over the BERT-CRF model. However, post-incremental training with an extensive corpus of traditional Chinese texts improves the performance of BERT for these two tasks, in relation to the BERT-CNN and BERT-CRF models (Tang et al., 2021).

Pre-trained Model The BERT model has gained significant prominence in various Chinese language processing tasks, including word segmentation, part-of-speech tagging, among others. Nonetheless, it is essential to note that BERT's pre-training primarily focuses on Simplified Chinese while SikuRoBERTa (Wang et al., 2022) on traditional Chinese texts. Consequently, SikuRoBERTa performs better in the situation of dealing with classical Chinese texts.

3. Our Method

In this section, we introduce our methods and model architectures.

The EvaHan2024 task encompasses two subtasks, i.e., the SS subtask and the PP subtask. Sentence boundaries are closely correlated with some punctuation marks, such as periods and exclamation marks. We call these punctuation marks *pause marks*. We call other punctuation marks *non-pause marks*. Table 1 lists the two types of punctuation marks.

Upon distinguishing the two types of punctuation marks, we propose to avoid the SS subtask and treat it as a part of the PP subtask. Moreover, we handle the two types of punctuation marks separately via sequence labeling.

3.1. Data Pre-processing

Figure 2 illustrates how to pre-process raw training data. The character sequence without punctuation marks composes an input sequence for the two independent sequence labeling models. The middle two rows give the tag sequences for the two models.

3.2. Two stages

The above pre-processing method leads to the problem of being unable to determine the order during post-processing when two CRFs predict marks at the same position. The high-frequency punctuation pairs in Table 2 illustrate that this problem cannot be avoided. We propose two methods to solve this problem.

Two-stage Method When we divide the punctuation points into two groups, we improve on

the post-processing method. We counts the frequency of different orders from the training set, and selects the order with higher frequency as the final result[1].

One-stage Method The one-stage method is to dropout the label grouping method and treat the PP task as one sequence labeling task instead of two. Specifically, we treat punctuation combinations that appear at the same position as one label. Moreover, some low-frequency labels can be mapped to high-frequency labels to simplify the label set.

We compare the performance of these two approaches in Table 5.

3.3. Models

The input sequence is defined as $S = \{c_0, c_1, ..., c_n\}$, where n represents sequence length and c_i denotes the i-th character of the sequence. The lowest embedding layer of the model utilizes SikuRoBERTa and character embedding.

The SikuRoBERTa output representation of character c_i is denoted as $\boldsymbol{e}_i^s$. The character embedding representation of character c_i is denoted as $\boldsymbol{e}_i^c$. The concatenation of $\boldsymbol{e}_i^s$ and $\boldsymbol{e}_i^c$ forms the embedding representation of character c_i, expressed as $\boldsymbol{e}_i$. The formulation of this representation is as follows:

$$\boldsymbol{e}_i = \boldsymbol{e}_i^s \oplus \boldsymbol{e}_i^c \tag{1}$$

After obtaining the embedding layer representation, it is encoded through three BiLSTM layers to derive the contextual representation.

$$\boldsymbol{R} = BiLSTMs(\boldsymbol{e}) \tag{2}$$

Within this framework, $\boldsymbol{e}$ signifies the embedding representation of the input sequence, while $\boldsymbol{R}$ is the context representation.

The final two layers consist of distinct MLP-CRF models. The MLP layer extracts information from the contextual representation and reduces the vector dimension to match the size of the label set.

$$\boldsymbol{S} = MLP(\boldsymbol{R}) \tag{3}$$

In this formula, $\boldsymbol{S}$ denotes the outputs of the MLP model.

Subsequently, the CRF layer calculates the CRF-loss during training and employs the Viterbi algorithm for inference purposes. The implementation of the CRF model is based on SuPar[2].

[1] In fact, we did not use this method when submitting the results, but rigidly placed all pause marks after non-pause marks. While this does not affect the calculated F1 score, we have modified this in the published code.

[2] Supar Github: https://github.com/yzhangcs/parser.

Data parameters	Numbers
Train set lines	263,091
Dev set lines	13,984
Chars	10,638
Max length	510
Window size	100
Tag combinations	160
Tag combinations of one-stage	72
Non-Pause tags	40
Pause tags	7

Table 3: Parameters after data processing.

Hyperparameters	Values
Dimension of SikuRoBERTa	100
Dimension of char embedding	100
Hidden dimension of BiLSTM	400
Dimension of MLP1	41
Dimension of MLP2	8
Learning rate of BiLSTM	2e-5
Learning rate of MLPs and CRFs	2e-4
Dropout ratios	0.33
Batch size	50

Table 4: Hyperparameters.

$$loss = crf\_loss(\boldsymbol{S}, y) \tag{4}$$

$$\hat{y} = Viterbi(\boldsymbol{S}) \tag{5}$$

In this context, y represents the ground truth while $\hat{y}$ signifies the prediction result.

4. Experiments

4.1. Data

In this task, the training data shared by EvaHan2024 originates from the *Siku Quanshu*, containing over 10 million characters. We designate 5% of the training data as provisional validation data for assessing the model's performance. Furthermore, in addition to this dataset, we employe the Xunzi-Qianwen-7B-CHAT to generate approximately 11,000 synthetic classical Chinese sentences. These generated data are utilized for both training and validation purposes.

The handling of long sequences poses a challenge. As these sequences represent a minority in the training data, they are typically truncated directly. For evaluation, we employs the parallel sliding window approach described in Tang et al. (2021) to manage using a fixed window size, without compromising efficiency and performance.

The parameters of processed dataset is shown in Table 3. The "Tag combinations" entry in Table 3 comprises a count of 160. This figure is the

Test A	Sentence Segmentation			Punctuation Prediction		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
Baseline	90.53	66.12	76.42	73.52	52.22	61.06
ChatGPT-3.5	83.81	59.85	69.83	63.90	43.88	52.03
Our Model (One-stage)	91.23	83.25	87.06	76.41	67.88	71.89
Our Model (Two-stage)	89.82	84.69	**87.18**	75.87	69.70	**72.66**
Test B	Sentence Segmentation			Punctuation Prediction		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
Baseline	95.28	87.17	91.04	79.25	72.09	75.50
Our Model (One-stage)	95.47	91.47	**93.43**	83.42	78.42	80.84
Our Model (Two-stage)	94.64	91.93	93.27	82.93	78.96	**80.89**

Table 5: Test set results. Test B is implemented on the Zuozhuan test set.

total number of punctuation combinations present in the dataset when they are not separately labeled. Upon labeling according to the classification method mentioned in Section 3, the size of the label set can be notably diminished to 40 and 7.

4.2. Results

The training hyperparameters are detailed in Table 4, with the Adam optimizer employed. The model training is conducted on an NVIDIA Tesla-V100-SXM2-32G GPU, utilizing a batch size of 50 which requires approximately 30G of memory per iteration. Each iteration takes 4.5 hours. Notably, it is observed that the model achieves optimal performance on the validation set in the 4th iteration.

In accordance with common practice, the evaluation of our model entails assessing its Precision (P), Recall (R), and F1 score. The results are presented in Table 5, it can be seen that the two-stage method performs better on the test set. The experimental results demonstrate that the task performance of our model vastly outperform the baseline model on both evaluation sets.

5. Discussion

In this task, our model shows robust performance, owing to several enhancements.

Firstly, we distinguish between non-pause and pause punctuation to simplify the process of sequence labeling. Secondly, introducing SikuRoBERTa and character embeddings into the model architecture to obtain embedding representations. In addition, we employ XunziALLM to generated classical Chinese writings for training and validation.

However, there are flaws in our approach.

Firstly, the two-stage method we mentioned in Section 3 is not elegant. Another idea is to train a binary classifier to determine the order. Secondly, an issue of incomplete data processing arises due to the expansive nature of the dataset and encoding difficulties associated with some traditional Chinese characters. Consequently, instances of missing characters or incomplete sentence are encountered. We treat these data as noise and remove them. Furthermore, we apply the rule-based method to correct the illegal punctuation marks within the dataset. It is acknowledged, however, that the efficacy of this correction method is limited. Thirdly, The BiLSTM layers process lengthy texts slowly, lengthen the training process. Moreover, The XunziALLM tool is not fully leveraged.

Acknowledgements

We thank organizers of the EvaHan2024 shared task for their help and hard work, all the anonymous reviewers for their valuable comments, and Jielin Chen for her help in improving the writing of this paper. This work was supported by National Natural Science Foundation of China (Grant No. 62176173 and 62336006), and a Project Funded by the Priority Academic Program Development (PAPD) of Jiangsu Higher Education Institutions.

6. References

Ning Cheng, Bin Li, Liming Xiao, Changwei Xu, Sijia Ge, Xingyue Hao, and Minxuan Feng. 2020. Integration of automatic sentence segmentation and lexical analysis of Ancient Chinese based on BiLSTM-CRF model. In *Proceedings of LT4HALA 2020 - 1st Workshop on Language Technologies for Historical and Ancient Languages*, pages 52–58, Marseille, France. European Language Resources Association (ELRA).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Renfen Hu, Shen Li, and Yuchen Zhu. 2021. Knowledge representation and sentence segmentation of ancient chinese based on deep language model. *Journal of Chinese Information Processing*, 35(4):8–15.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

John Lafferty, Andrew McCallum, Fernando Pereira, et al. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Icml*, volume 1, page 3. Williamstown, MA.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Xuemei Tang, Qi Su, Jun Wang, Yuhang Chen, and Hao Yang. 2021. Automatic traditional Ancient Chinese texts segmentation and punctuation based on pre-training language model. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 678–688, Huhhot, China. Chinese Information Processing Society of China.

Dongbo Wang, Chang Liu, Zihe Zhu, Jiangfeng Liu, Haotian Hu, Si Shen, and Bin Li. 2022. Construction and application of pre-trained models of siku quanshu in orientation to digital humanities. *Library Tribune*, 42(6):31–43.

Han Xu, Wang Hongsu, Zhang Sanqian, Fu Qunchao, and Liu Jun. 2019. Sentence segmentation for classical chinese based on lstm with radical embedding. *The Journal of China Universities of Posts and Telecommunications*, 26(2):1.