Shanduan: Sentence Segmentation for Classical Chinese Based on BERT

Xiaodi Yuan
2019012382
Tsinghua University
yuanxd19@mails.tsinghua.edu.cn

Kangbo Lyu
2019012378
Tsinghua University
lkb19@mails.tsinghua.edu.cn

1 Introduction

Classical Chinese (文言文) is known for its lack of a punctuation system. Although analogs of punctuation do exist in a portion of literature, those marks vary from book to book and do not form a unified system. This special phenomenon makes reading more difficult for a wide range of readers, from children in ancient times (who had to learn a technique called "Judou (句读, punctuation)" first to read) to ordinary modern readers.

In this paper, we propose a BERT-based model called Shanduan (善断, namely a model good at segmentation) to automatically segment and punctuate classical Chinese texts.

Our code is released at `https://github.com/Rabbit-Hu/shanduan`.

2 Related Work

2.1 Statistical Models with Manual Features

Huang and Hou [1] made one of the first attempts to automatic sentence segmentation in classical Chinese using a rule-based method with manually designed features. Chen et al. [2] used n-gram method to further integrate the context information. The performances of these methods are very limited.

Zhang et al. [3] made the first breakthrough. They proposed a model using conditional random field (CRF) as the statistical model with two kinds of manual features, the mutual information and the t-test difference. Besides, some variants made slight improvements by adding more manual features [4] [5].

Manual Features One of the features used in this model is the mutual information between two adjacent characters s_i, s_{i+1}, and the other is the t-test difference among three consecutive characters s_{i-1}, s_i, s_{i+1}, where the probability is estimated by the frequency, and σ^2 denotes the variant.

$$I(s_i, s_{i+1}) = \log \frac{p(s_i, s_{i+1})}{p(s_i)p(s_{i+1})}$$

$$t(s_{i-1}, s_i, s_{i+1}) = \frac{p(s_{i+1}|s_i) - p(s_i|s_{i-1})}{\sqrt{\sigma^2(p(s_{i+1}|s_i)) + \sigma^2(p(s_i|s_{i-1}))}}$$

Conditional Random Field Conditional random field (CRF) is proposed by Lafferty *et al.* [6] to make full use of manual features to tag sequence and outperforms other linear statistical models such as the hidden Markov model (HMM) and the maximum entropy Markov model (MEMM). For given feature extractors f, CRF model computes the score of

the labels **y** on given inputs **s** with linear combination of features with learnable weights.

$$score((y_1, y_2, \ldots, y_n)|(s_1, s_2, \ldots, s_n)) = \sum_{j=1}^{m} \sum_{i=1}^{n} w_j f_j(s, i, l_{i-1}, l_i)$$

By setting the log of softmax of scores as the loss, one can train the weights **w** by back-propagation. At the inference stage, the goal is to maximize the score on given inputs **s** and fixed weights. An efficient algorithm is to find a Viterbi path by dynamic programming. [6]

This model does not require many computational resources. Jiayan(甲言)[1], a Python NLP tool for classical Chinese, uses this model for sentence segmentation due to its competitively high accuracy and efficiency, which can be regarded as a simple baseline.

2.2 RNN with CRF Layers

The accuracy of CRF models highly relies on the feature extractors. By replacing the manually designed feature extractors with recurrent neural networks (RNN), which can learn the features automatically, one can obtain higher accuracy.

Long Short-Term Memory With additional input gates, forget gates, and output gates, the long short-term memory model (LSTM) is one of the most widely used RNN. Cheng et al. [7] investigated a bidirectional LSTM model with additional CRF layers, leading to significant performance gains.

Gated Recurrent Unit The gated recurrent neural network, which consists of the gated recurrent units (GRU), is a successful variant of LSTM in sequence modeling, proposed by Cho et al. [8]. Wang et al. [9] employed the GRU with additional CRF layers to make more progress on sentence segmentation in classical Chinese.

Ablation studies in both models [7] [9] show that the CRF layers are essential in this task since RNN with additional CRF layers always outperforms the corresponding RNN with a vanilla softmax layer.

2.3 Pre-trained Deep Transformers

Vaswani et al. [10] proposed the Transformer model which dominated the field of sequence modeling afterward. It consists of an encoder and decoder, where the scaled dot-product attention mechanism is firstly introduced and plays an important role in improving performance. We are expecting another breakthrough on sentence segmentation in classical Chinese by Transformers.

BERT Devlin et al. [11] proposed a pre-training pipeline for deep bidirectional transformers on several predicting tasks. As a language model, such a pre-trained BERT model can handle multiple tasks by fine-tuning on smaller datasets. Yu et al. [12] investigated the performance of a BERT model on the segmentation task of classical Chinese, which significantly outperforms the RNN-based methods. Also, ablation studies show that for a BERT model, the CRF layers cannot increase the accuracy and therefore are not required. [12]

RoBERTa Liu et al. [13] introduced several simple modifications on the pre-training pipeline of a BERT model, known as the advanced techniques called RoBERTa. The essential modifications include but are not limited to dynamic masking and also some changes in the objectives and hyperparameters. This is mainly what our model is based on, yet no application of this model on sentence segmentation of classical Chinese has been reported.

3 Problem Setup

We formalize both problems, sentence segmentation and punctuation, as token classification tasks. In the segmentation task, each character is labeled "1" if it is located before a break

[1] Jiayan `https://github.com/jiaeyan/Jiayan`.

and “0” if it is not, forming a binary classification task. Similarly, the punctuation is a multi-class classification task: instead of “1”, each character is labeled according to the type of the punctuation mark that follows it. Our model is supposed to take an unpunctuated sentence (with Chinese characters only) and predict the label for each character.

To simplify the problem, we further define “the punctuation marks of interest” as: “， ”, “。”, “、”, “? ”, “! ”, “: ”, “; ”. All marks besides those seven symbols, including quote marks and parentheses, are removed from the texts since the removal of these marks avoids the continuous occurrence of punctuation marks and does not affect the general meaning of each sentence. Therefore, the punctuation task only involves 8 labels (label “0” means there is no punctuation).

4 Our Method

4.1 Dataset

4.1.1 Pre-training corpus.

Our model is based on a pre-trained RoBERTa model named GuwenBERT[2]. It’s trained on a large corpus of ancient Chinese books called “DaizhiGe (殆知阁)”[3]. This dataset, with a size of 4.9G, contains nearly all well-known works that are handed down in Chinese history. However, most of the text files have not been proofread, thus there are many typos, garbled codes, and the formats of each file are not consistent. For our task, the most significant flaw of this dataset is the absence of punctuation marks in some of the books and the different rules of punctuation that other books follow.

4.1.2 Fine-tuning dataset.

We fine-tune our model on a smaller but cleaner dataset. The text files are from Chinese Text Project (hereinafter referred to as CText) [4], an online library that provides reliable digital ancient Chinese books. The size of this library is much more limited than the previous dataset, DaizhiGe, but it has a higher quality: thanks to the proofreading by contributors over the world, the texts have few typos and a consistent rule of punctuation.

We downloaded four books from CText: *Records of the Grand Historian* (《史记》), *Book of Han* (《汉书》), *Book of the Later Han* (《后汉书》), and *Records of the Three Kingdoms* (《三国志》). All the books are “official history books (正史)”. All the four books are also contained in DaizhiGe, but for the last book, *Records of the Three Kingdoms*, DaizhiGe only has an unpunctuated version. Considering the extreme difficulty of finding reliably punctuated texts that are surely not in DaizhiGe, we decide to use this book (2.5M) for evaluation on which we measure the performance of our model and the baseline, Jiayan; and we use the first three books (6.7M in total) for training.

To form a dataset for fine-tuning a segmentation model, after removing all symbols except Chinese characters and the punctuation marks of interest, the texts are divided into sentences according to five punctuation marks “。”, “? ”, “! ”, “: ”, “; ”. Adjacent sentences are merged into text snippets of length less than 256.

4.2 Model

Compared to RNN models [7] for segmentation that require additional CRF layers, our model is extremely simple. We are surprised to find that with only one linear layer stacked after the output layer of RoBERTa, the model can achieve satisfactory performance.

[2]GuwenBERT `https://github.com/ethan-yt/guwenbert`.

[3]The official website: `http://www.daizhige.org/`; all the files can be downloaded from this GitHub repository: `https://github.com/garychowcmu/daizhigev20`.

[4]Chinese Text Project (中国哲学书电子化计划) `https://ctext.org/`.

4.3 Training

To fine-tune the model, we use cross entropy for the loss function and AdamW [14] for the optimizer. The learning rate is warmed up from 0 to 10^{-5} linearly in the first 5 epochs and then decays to 0 in the following 45 epochs. We use early stopping to alleviate overfitting (the best validation performance is usually reached within the first 15 epochs).

It is noteworthy that with the help of a pre-trained model, the fine-tuning process is really fast. Each epoch takes around 3.5 minutes on a single GeForce RTX 2080 Ti, thus the best model can be obtained within one hour.

5 Evaluation

5.1 Metrics

It should be noted that both the sentence segmentation task and the punctuation prediction task have extremely unbalanced labels. For example, as is shown in Table 1, a model that predicts "0" for all characters will achieve an accuracy of 80.3% on the segmentation task. Moreover, Table 2 shows that the imbalance within the "punctuation" category is also severe, as most punctuation marks are commas. Therefore, accuracy is not a suitable metric for our task, and we should take both precision and recall into consideration to compare model performances.

Label	non-punctuation	punctuation
Ratio	80.3%	19.7%

Table 1: Label distribution in the sentence segmentation task.

Label	non-punctuation	,	。	、	?	!	:	;
Ratio	80.3%	12.3%	4.9%	0.9%	0.3%	0.2%	1.1%	0.2%

Table 2: Label distribution in the punctuation prediction task.

For segmentation (binary classification task), the solution is simply choosing the $F1$ score for the main metric. However, things are a bit more complicated for punctuation (multi-class classification task).

F1 defined in a previous work. In a previous work on punctuation prediction, Che et al. [15] re-defined precision, recall and F1 by:

$$Precision = \frac{\#\text{ Correctly predicted punctuation marks}}{\#\text{ All predicted punctuation marks}} \tag{1}$$

$$Recall = \frac{\#\text{ Correctly predicted punctuation marks}}{\#\text{ All expected punctuation marks}} \tag{2}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3}$$

This metric addresses the imbalance between the two categories of "punctuation" and "non-punctuation", but it ignores the imbalance among different punctuation marks and considers all kinds of misclassifications within the "punctuation" category are considered equivalent. For example, the ratio of commas ("，") and semicolons ("；") is about 50 : 1 in our corpus. If model A learns nothing about semicolons and misclassifies all semicolons into periods, while model B masters semicolons but happens to misclassify 2% commas into periods, the two models will have equal $F1$ scores according to this metric definition. However, we consider that model B outperforms model A because it is a fatal weakness for A to know nothing about semicolons.

Metric		Precision		Recall		F1	
Model		Jiayan	Shanduan	Jiayan	Shanduan	Jiayan	Shanduan
Punc-tuation	none	96.6	**99.1**	96.7	**98.3**	96.6	**98.7**
	,	61.0	**84.2**	86.9	**89.0**	71.7	**86.5**
	。	30.9	**79.8**	4.8	**82.8**	8.3	**81.3**
	、	80.7	**86.8**	2.3	**74.7**	4.4	**80.3**
	?	58.9	**76.4**	**70.7**	63.3	64.3	**69.2**
	!	48.0	**48.7**	23.1	**57.7**	31.2	**52.9**
	:	93.8	**94.4**	85.6	**94.9**	89.5	**94.6**
	;	33.8	**39.9**	5.0	**19.4**	8.7	**26.1**
	mean	63.0	**76.1**	46.9	**72.5**	46.8	**73.7**
Segmentation		86.4	**94.7**	86.0	**95.4**	86.2	**95.0**

Table 3: Experimental results of Jiayan (the baseline) and Shanduan (our model) on the validation set. The last row (named "mean") compares $mPrecision$, $mRecall$, and $mF1$ defined as in Section 5.1.

Our metric. We choose to evaluate punctuation models by the mean $F1$ score (hereinafter referred to as mF1) over all the eight classes, that is, for $0 \leq i \leq C-1$ (C = # classes = 8):

$$Precision_i = \frac{TP_i}{TP_i + FP_i}, \quad Recall_i = \frac{TP_i}{TP_i + FN_i}, \tag{4}$$

$$F1_i = \frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i}, \quad mF1 = \frac{1}{C}\sum_{i=1}^{C} F1_i. \tag{5}$$

To further analyze the behavior patterns of punctuation models, we define mean precision and mean recall:

$$mPrecision = \frac{1}{C}\sum_{i=1}^{C} Precision_i, \quad mRecall = \frac{1}{C}\sum_{i=1}^{C} Recall_i. \tag{6}$$

For the example of periods and semicolons mentioned above, the absence of semicolons in the prediction of model A will seriously reduce $F1_{\text{semicolon}}$ and hence reduce $mF1$ by up to $\frac{1}{8}$, which noticeably reflects the defect of model A.

5.2 Results

Table 3 shows the evaluation results of the baseline, CRF model Jiayan (see Section 2.1), and our model Shanduan. For the baseline model, we load the pre-trained weights provided on the official GitHub repository[5] of Jiayan, which is also trained on DaizhiGe corpus (see Section 4.1.1)).

The table shows that our model outperforms Jiayan in almost all the evaluation indicators. In particular, our model greatly improves the recall of those punctuation marks that appears less often, which results in a significant improvement in $mF1$. For example, the highlighted 2×2 submatrix in Figure 2 shows that Jiayan tends to predict commas where the ground truth is period, while the misclassifications of our model are less and more balanced.

5.3 Examples

To intuitively compare the performance of our model and the baseline, we select an article that is not contained in any training corpus and is particularly **hard** to understand even for human beings. The outputs of the two models are listed in Table 4.

[5]Jiayan `https://github.com/jiaeyan/Jiayan`.

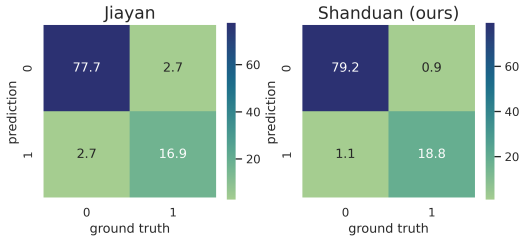

Figure 1: Confusion matrices of Shanduan and Jiayan on the sentence segmentation task.

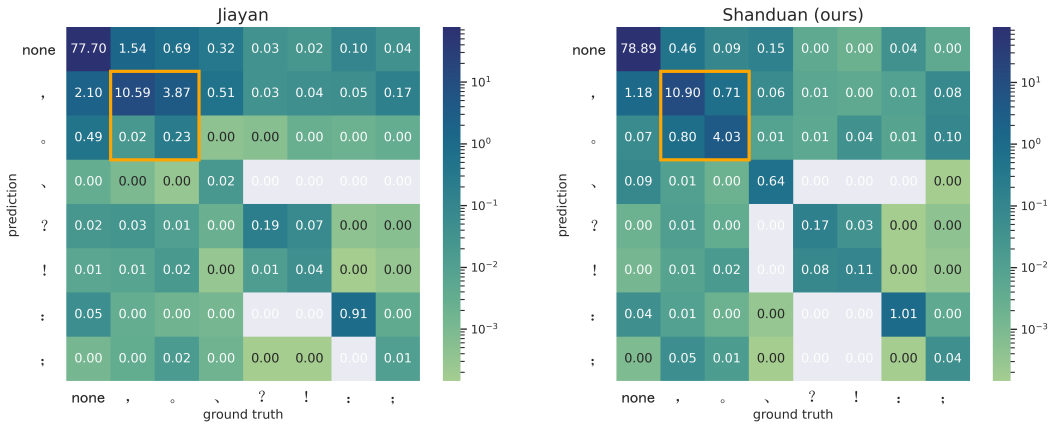

Figure 2: Confusion matrices of Shanduan and Jiayan on the punctuation prediction task.

Ground truth	石室诗士施氏，嗜狮，誓食十狮。施氏时时适市视狮。十时，适十狮适市。是时，适施氏适市。施氏视是十狮，恃矢势，使是十狮逝世。氏拾是十狮尸，适石室。石室湿，氏使侍拭石室。石室拭，施氏始试食是十狮尸。食时，始识是十狮尸，实十石狮尸。试释是事。
Jiayan	石室诗士，施氏嗜狮，誓食十狮，施氏时时适市，视狮十时适十狮，适市，是时适施氏，适市施氏，视是十狮，恃矢势使，是十狮逝，世氏拾是十，狮尸适石室，石室湿氏，使侍拭石室，石室拭施氏，始试食是十狮尸，食时始识是十，狮尸实十石，狮尸试释是事。
Shanduan (ours)	石室诗士施氏嗜狮，誓食十狮。施氏时时适市，视狮十。时，适十狮适市，是时适施氏适市，施氏视是十狮，恃矢势，使是十狮逝世氏拾是十狮尸适石室石室，湿氏使侍拭石室石室拭，施氏，始试食是十狮尸，食时始识是十狮尸，实十石狮尸，试释是事。

Table 4: A hard example, *Lion-Eating Poet in the Stone Den* written by the Chinese linguist Yuen Ren Chao in the 1930s.

6 Conclusion and Future Work

In this paper, we proposed Shanduan, a simple model based on pre-trained RoBERTa that predicts segmentation and punctuation for unpunctuated classical Chinese texts. Compared to classical statistical algorithm Jiayan, our model significantly improved recall of low-frequency labels and hence the mean $F1$ score.

We also created a new dataset to finetune the model and analyzed the metrics to reliably evaluate the performances of different segmentation or punctuation models.

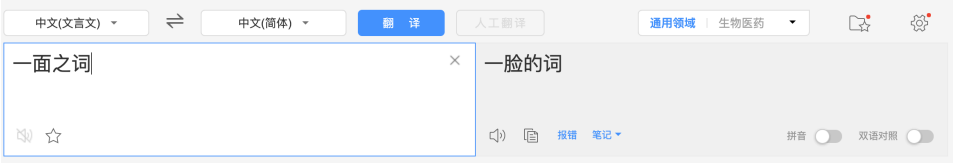

Figure 3: Baidu Translator translates "the statement of only one of the parties" word for word into "a face of words".

Currently, tools for classical Chinese studies are not "intelligent" enough. For example, we failed to find any satisfactory classical Chinese translators on the Internet: the one provided by Baidu seems to be a verbatim translator, as is shown in Figure 3. Therefore, we hope to further explore the great potential of pre-trained BERT models on classical Chinese analysis in the future.

References

[1] Jiannian Huang and Hanqing Hou. On sentence segmentation and punctuation model for ancient books on agriculture. *Journal of Chinese Information Processing*, 22(4):31–38, 2008.

[2] Tianying Chen, Rong Chen, Lulu Pan, Hongjun Li, and Zhonghua Yu. Archaic chinese punctuating sentences based on context n-gram model. *Jisuanji Gongcheng / Computer Engineering*, 32(3):192–193, 2007.

[3] Kaixu Zhang, Yunqing Xia, and Hang Yu. Crf-based approach to sentence segmentation and punctuation for ancient chinese prose. *Journal of Tsinghua University (Science and Technology)*, 10, 2009.

[4] Hen-Hsen Huang, Chuen-Tsai Sun, and Hsin-Hsi Chen. Classical chinese sentence segmentation. In *CIPS-SIGHAN joint conference on Chinese language processing*, 2010.

[5] M. Shi, L. I. Bin, and X. Chen. Crf based research on a unified approach to word segmentation and pos tagging for pre-qin chinese. *Journal of Chinese Information Processing*, 2010.

[6] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[7] Ning Cheng, Bin Li, Liming Xiao, Changwei Xu, Sijia Ge, Xingyue Hao, and Minxuan Feng. 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, 2020.

[8] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[9] Boli Wang, Xiaodong Shi, Zhixing Tan, Yidong Chen, and Weili Wang. A sentence segmentation method for ancient chinese texts based on nnlm. In *Workshop on Chinese Lexical Semantics*, pages 387–396. Springer, 2016.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] Jingsong Yu, Yi Wei, and Yongwei Zhang. automatic ancient chinese texts segmentation based on bert. *Journal of Chinese Information Processing*, 33(11):57–63, 2019.

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

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[15] Xiaoyin Che, Cheng Wang, Haojin Yang, and Christoph Meinel. Punctuation prediction for unsegmented transcript based on word vector. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 654–658, 2016.