SPEADO: Segmentation and Punctuation for Ancient Chinese Texts via Example Augmentation and Decoding Optimization

Xia Tian[1], **Yu Kai**[2], **Yu Qianrong**[1], **Peng Xinran**[1]
[1]School of Information Resource Management, Renmin University of China, Beijing, China
[2]Shanghai Midu Technology Co., Ltd., Shanghai, China
[1]{xiat, yuqianrong77, pengxinran166}@ruc.edu.cn
[2]yukai@midu.com

Abstract

The SPEADO model for sentence segmentation and punctuation tasks in ancient Chinese texts is proposed, which incorporates text chunking and MinHash indexing techniques to realise example argumentation. Additionally, decoding optimization strategies are introduced to direct the attention of the LLM model towards punctuation errors and address the issue of uncontrollable output. Experimental results show that the F_1 score of the proposed method exceeds the baseline model by 14.18%, indicating a significant improvement in performance.

Keywords: Sentence segmentation and punctuation, Ancient Chinese texts, Large Language Models

1. Introduction

Ancient texts, a crucial component of Chinese culture, are abundant in historical, cultural, and ideological value. However, their distinctive ancient writing style often lacks explicit sentence breaks and punctuation, rendering them difficult to read and comprehend. While traditional manual annotation methods can provide assistance, the vast quantity of ancient Chinese texts makes manual processing inefficient and costly, limiting digital processing and large-scale research efforts. Fortunately, with the advancements in Large Language Model (LLM) technologies, it has become more feasible to efficiently tackle this challenge.

This task can be regarded as a generation task, involving the conversion of unpunctuated sentences into punctuated ones. LLMs, such as XunziALLM, have demonstrated remarkable fundamental capabilities in this regard. We propose an augmentation method inspired by human learning through examples, coupled with decoding strategies to enhance task focus and output control. Fine-tuning with LoRA enables our SPEADO model to learn the skill of punctuating ancient Chinese texts, significantly improving performance over baseline models.

2. Related Work

The sentence segmentation and punctuation tasks are crucial for parsing the meaning of Ancient Chinese texts. Research on automated annotation methods for these tasks can be categorized into several stages: rule-based, statistical, and deep learning approaches. Early attempts to these tasks primarily relied on rule-based systems (Huang and Hou, 2008). While effective in some cases, rule-based approaches often struggled with ambiguous syntactic structures and variations in writing styles. Moreover, maintaining and updating rule sets proved to be labor-intensive and prone to errors. Therefore, researchers started exploring natural language statistical modeling, particularly the development of N-gram models that capitalized on contextual features to predict sentence boundaries (Cheng et al., 2007).

With the development of the field of deep learning and the advancement of sentence segmentation and punctuation tasks in Acient Chinese, models such as BERT, LSTM/BiLSTM, and CRF (Yu et al., 2019; Wang et al., 2021) have been proven to exhibit strong performance in there.Subsequently, researchers have shifted their focus towards optimizing these network architecture.

Some researchers have focused on optimizing pre-trained models and, based on large-scale Classical Chinese datasets, have respectively trained pre-trained models tailored for Classical Chinese, namely BERT_guwen and SikuBERT. Some scholars have incorporated fine-grained textual knowledge and adjusted the model structure using CNN and BiLSTM, proposing the BBiCC-EK (BBiC-CNN-External Knowledge) model (Li et al., 2023). Moreover, Considering that separating punctuation and sentence segmentation in classical texts into two sequential tasks may lead to error propagation, some studies treat the segmentation and punctuation of ancient texts as a joint task (Yuan et al., 2022).

Notable Chinese LLMs include Baidu's Ernie (Yu et al., 2021) and Alibaba Cloud's Qwen (Jinze et al., 2023), demonstrating excellent language understanding and generation abilities. For Acient Chinese, Nanjing Agricultural University and Zhonghua Book Company's joint efforts have produced a series of LLMs specialized in processing classical texts, named as the XunziALLM. These models exhibit impressive performance in handling Ancient Chinese textual information. Leveraging XunziALLM as base model and optimizing it for the joint task like punctuation and sentence segmentation in classical texts seems like a promising choice.

3. Method

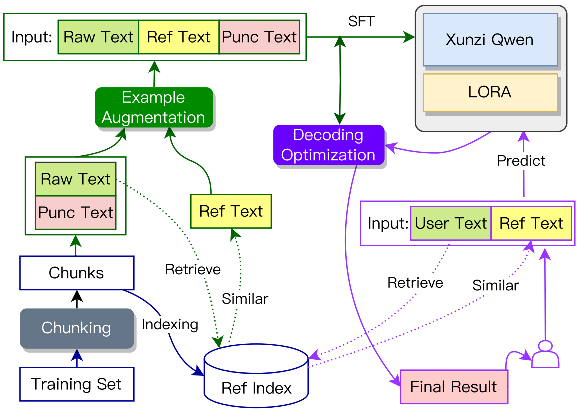

Figure 1: Architecture of SPEADO.

Recognizing that the punctuation in ancient Chinese texts encodes crucial information for sentence segmentation, we have merged the tasks of automatic sentence segmentation and punctuation, introducing an integrated approach named SPEADO. SPEADO, an acronym for sentence segmentation and punctuation via example augmentation and decoding optimization, offers a comprehensive solution to the challenges posed by these tasks. As depicted in Figure 1, SPEADO comprises three core modules: text chunking, example augmentation, and decoding optimization.

3.1. Chunking Process

Because the lengths of the samples in the training data vary significantly, directly utilizing each row as a standalone sample for input into the training network can result in truncation issues and hinder the training speed. To mitigate this issue, we have employed a sliding window mechanism that divides the training dataset into chunks, thereby enabling the generation of additional training samples.

As depicted in Figure 2, let us consider the raw text X comprising of m sentences, denoted as $X = x_1, \cdots, x_m$. We proceed to transform X into a series of length-constrained chunks, designated as $C = \{c_1, c_2, ..., c_n\}$. In the process of conversion, we traverse X from the left to the right, iteratively generating each chunk c_i in the following manner:

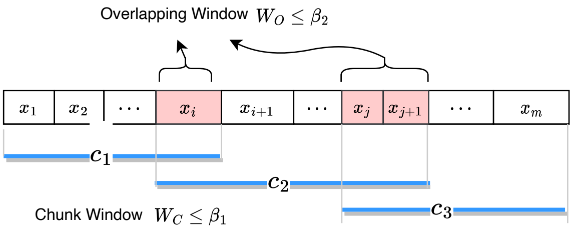

Figure 2: Illustration of Overlapping Chunks.

1. The chunk c_i starts at x_a and ends at some x_b, where $a \le b \le m$, and the initial value of a is 1;
2. Find the largest value b that satisfies $\sum_{k=a}^{b} len(x_k) \le \beta_1$;
3. Set the chunk $c_i = \{x_a, x_{a+1}, \cdots, x_b\}$;
4. Starting from x_b, backtrack to find the smallest value $c \ge 1$ that satisfies $\sum_{k=c}^{b} len(x_k) \le \beta_2$;
5. Set $a = b$, repeat step 1, and obtain all valid chunks in turn.

Here, β_1 represents the maximum value for the window size of a text chunk, while β_2 denotes the maximum value for the overlapping window.

3.2. Example Augmentation

When humans tackle tasks, the provision of pertinent reference information greatly aids in their resolution. Drawing inspiration from this, we incorporate correct reference examples with the original text during the training process of our model for automatic punctuation of ancient texts. This allows LLM to refer to relevant information to better perform the punctuation task.

To achieve this, we pre-construct a MinHash index for the text chunks obtained in the previous step. The utilization of MinHash, rather than text embedding techniques, stems from the fact that semantic embedding models exhibit limited effectiveness in comprehending ancient Chinese texts. Consequently, MinHash is more adept at retrieving and matching character similarity. After performing the MinHash operation, a reference index database (i.e., Ref Index) is created for the text chunks, enabling us to retrieve examples for reference purposes.

For a manually punctuated text chunk c, we establish the punctuated text as the gold standard and proceed to strip it of all punctuation, yielding the raw text that requires punctuation prediction. Subsequently, we compute the MinHash value of the raw text and utilize it to retrieve a similar text from the refdb, designated as the reference text. The raw text, reference text, and the original punctuated text are then merged according to a prescribed prompt template, culminating in a comprehensive training data input tailored for supervised fine-tuning within LLM.

3.3. Decoding Optimization

After fine-tuning, the LLM still demonstrates unpredictable behavior during prediction, such as generating characters that are neither punctuation nor original text, and reproducing entire sentences without any modification. To tackle this issue, we have incorporated three types of optimization techniques during the model decoding process.

Firstly, we refined the loss function during training to prioritize punctuation errors. Whenever the punctuation placement in the output is inaccurate, we enhance the original loss value using a factor, λ, set to 0.05 in our experiments.

Secondly, during prediction, we imposed a decoding constraint that confines the next predicted character to either the original input character or punctuation marks. Subsequently, we selected the character with the highest logits value from this constrained set as the final prediction, effectively addressing issues pertaining to inconsistent output characters.

Lastly, we utilized a voting mechanism for unchanged sentences after prediction. We trained three models using LoRA fine-tuning based on the Xunzi-Qwen-7B. These included SPEADO-A (standard LoRA fine-tuning), SPEADO-B (with example augmentation), and SPEADO-C (with loss adjustment). These three models form the expert model, which votes on unmodified sentences and re-selects the prediction results.

4. Experiments

4.1. Data and Evaluation Metrics

We employed the dataset provided by the EvaHan2024 organizers for both training and testing. The training set included 10 million characters extracted from the Complete Library of Four Branches. The test sets comprise A and B, where the former refers to the data released initially, and the latter refers to the Zuozhuan data released the second time.

In the model validation phase, we randomly sampled 10,000 lines of text without replacement from the training set to create a validation set, reserving the remaining data for model training, to observe the varying impacts of different factors on the model. In the final stage, we utilized all the data as the training set to predict on the test sets. The prediction results were then submitted to the organizers for metric calculations. Table 1 presents the detailed statistical information of the dataset.

Dataset	Samples	Max Len	Avg Len
Training Set	254,360	29,907	93
Validation Set	10,000	3,546	116
Test Set - A	412	1,569	122
Test Set - B	3,319	656	59

Table 1: Statistics of EvaHan2024 dataset.

Table 1 reveals a considerable disparity in the average length of samples, with the longest sentence in the training set spanning 29,907 characters while averaging just 93. To mitigate this and enhance our dataset, we divided the raw text into chunks using parameters $\beta_1 = 256$ and $\beta_2 = 128$. This approach not only augmented our sample size but also addressed the challenge of excessively long inputs.

Following the convention of Seg and Punc tagging, we use Precision (P), Recall (R), and F1 Score as the evaluation metrics for all experiments. All the results are presented in percentages (%).

4.2. Implementation Details

For all experiments, we utilize the Xunzi-Qwen-7B as the backbone, employing a learning rate of $1 \times 10E - 5$ for the PLM. We adopt AdamW as the optimizer and WarmupDecayLR as the scheduler. Each GPU is assigned a micro-batch size of 2 for training. All our experiments are conducted on A100 GPUs, requiring approximately 60GiB of GPU memory and taking around 12 hours to achieve optimal performance.

4.3. Results

We compared five different methods on the validation set, and the results are presented in Table 2.

In Table 2, M_1 directly utilizes the Xunzi-Qwen-7B-CHAT model, revealing that the instructed LLM already possesses a certain level of ability in segmentation and punctuation task. M_2 fine-tunes the Xunzi-Qwen-7B base model using LoRA, significantly improving the performance compared to the chat model, emphasizing the necessity of secondary training for specific tasks. M_3 corresponds to the results after fine-tuning with the Xunzi-Baichuan-7B base model, aimed at verifying the differences between various base models. The data indicates that Xunzi-Qwen-7B slightly outperforms Xunzi-Baichuan-7B in this task, leading to our choice of Xunzi-Qwen-7B as our base model.

M_4 investigates the impact of weighting the loss related to punctuation positions during training. We observed a slight decline in certain metrics. Upon analysis, we found that the model's sensitivity to

ID	Base Model	Tuning Method	Seg			Punc		
			P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)
M_1	Xunzi-Qwen-7B-CHAT	—	69.60	76.61	72.94	55.02	60.56	57.65
M_2	Xunzi-Qwen-7B	LoRA	75.10	81.57	78.20	62.22	67.57	64.78
M_3	Xunzi-Baichuan-7B	LoRA	75.28	80.80	77.95	62.33	66.90	64.54
M_4	Xunzi-Qwen-7B	LoRA	74.35	80.80	77.40	61.17	66.48	63.71
M_5	Xunzi-Qwen-7B	LoRA	**75.93**	**81.90**	**78.80**	**62.82**	**67.75**	**65.19**

Table 2: Comparison of different methods on the validation set.

Test Sets	Model	Seg			Punc		
		P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)
Test Set - A	Xunzi-Qwen-7B-CHAT	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
	SPEADO	**90.99**	**85.99**	**88.42**	**78.75**	**72.02**	**75.24**
Test Set - B	Xunzi-Qwen-7B-CHAT	95.28	87.17	91.04	79.25	72.09	75.50
	SPEADO	**95.05**	**90.05**	**92.48**	**82.92**	**77.30**	**80.01**

Table 3: Comparison of various methods on the test sets. The asterisk (*) signifies that the EvaHan2024 organizer supplied the test results.

punctuation positions increased, resulting in more precise and nuanced punctuation usage. However, this enhanced sensitivity also led to the generation of redundant punctuation marks. Further exploration is needed to retain the model's stronger correction abilities while suppressing excessive modifications. M_5 introduces an example augmentation technique, which enables the model to better tackle the task by providing similar reference examples. This method demonstrated significant effectiveness.

During the testing phase, we introduced a comprehensive decoding enhancement strategy, employing M_2, M_4 and M_5 as expert model A, B, and C, respectively, to form the complete SPEADO model for prediction. As shown in Table 3, it is evident that SPEADO significantly improves the effectiveness of the tasks compared to the baseline model, Xunzi-Qwen-7B-CHAT.

5. Conclusion

Drawing from the previously mentioned research, it becomes apparent that the combination of example augmentation and decoding optimization can greatly enhance the abilities of LLMs in understanding and addressing tasks related to sentence segmentation and punctuation in ancient Chinese texts. This approach effectively tackles the challenge of uncontrollable output that is typically inherent in LLMs. Furthermore, training on the LLM-base has proven to be a more efficient and targeted means of achieving specific task objectives, surpassing the performance of the LLM-chat version.

6. Acknowledgements

This research is supported by the the National Social Science Fund of China (22BTQ068, 20&ZD260) and the Research Funds of Renmin University of China (22XNQT45).

7. References

Tianying Cheng, Rong Cheng, Lulu Pan, Hongjun Li, and Zhonghua Yu. 2007. Archaic chinese punctuating sentences based on context n-gram model. *Computer Engineering*, (03):192–193+196.

Jiannian Huang and Hanqing Hou. 2008. Review and trend of researches on ancient chinese character information processing. *Journal of Chinese Information Processing*, 22(4):31–38.

Bai Jinze, Bai Shuai, Chu Yunfei, and et al. 2023. Qwen technical report. Technical report.

Peiqi Li, Hao Wang, Qiutong Ren, and Tao Fan. 2023. Study of antiquarian punctuation recognition methods incorporating semantic enhancement with structural properties. *Journal of the China Society for Scientific and Technical Information*, (02):150–163.

Qian Wang, Dongbo Wang, Bing Li, and Chao Xu. 2021. Deep learning based automatic sentence segmentation and punctuation model for massive classical chinese literature. *Data Analysis and Knowledge Discovery*, (03):25–34.

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

Sun Yu, Wang Shuohuan, Feng Shikun, and et al. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint*.

Yiguo Yuan, Bing Li, Minxuan Feng, Sheng He, and Dongbo Wang. 2022. A joint model of automatic sentence segmentation and punctuation for ancient classical texts based on deep learning. *Library and Information Service*, (22):134–141.