Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA 2022), pages 164–168
Language Resources and Evaluation Conference (LREC 2022), Marseille, 25 June 2022

The Uncertainty-based Retrieval Framework for Ancient Chinese CWS and POS

Pengyu Wang, Zhichen Ren
Fudan University, Tongji University
220 Handan Road Shanghai, China
4800 Caoan Highway, Shanghai, China
wpyjihuai@gmail.com, 1850091@tongji.edu.cn

Abstract

Automatic analysis for modern Chinese has greatly improved the accuracy of text mining in related fields, but the study of ancient Chinese is still relatively rare. Ancient text division and lexical annotation are important parts of classical literature comprehension, and previous studies have tried to construct auxiliary dictionary and other fused knowledge to improve the performance. In this paper, we propose a framework for ancient Chinese Word Segmentation and Part-of-Speech Tagging that makes a twofold effort: on the one hand, we try to capture the wordhood semantics; on the other hand, we re-predict the uncertain samples of baseline model by introducing external knowledge. The performance of our architecture outperforms pre-trained BERT with CRF and existing tools such as Jiayan.

Keywords: Bigram Features, Uncertainty Sampling, Knowledge Retrieval

1. Introduction

Chinese Word Segmentation (CWS) and Part-of-Speech (POS) Tagging are two important tasks of natural language processing. With the rapid development of deep learning and pre-trained models, the performance of CWS and POS Tagging increased significantly. A simple model using pre-trained BERT and conditional random field (CRF) can reach a high accuracy. Since words are the most common components in a Chinese sentence and words can cause ambiguity, structures that can capture word information have been used in these tasks to get better performance.

Lexicon-based methods have been widely used in CWS, Chinese POS tagging and NER tasks to capture wordhood information (Yang et al., 2018; Li et al., 2020). These methods can leverage semantic information of words and improve model performance. However, lexicon-based methods have several drawbacks. One of the most severe problems is that they depend heavily on the quality of lexicons. Unfortunately, building an ancient Chinese lexicon is more difficult than building a modern Chinese lexicon, since there are few ancient Chinese corpus, and words from different corpus are different.

Further, sentences in ancient Chinese are always shorter than sentences in Chinese, which means words in ancient Chinese have a richer meaning and can cause misunderstanding or wrong classification.

The two problems mentioned above make ancient Chinese CWS and POS Tagging a harder problem. In our model, we combine bigram features with BERT to capture wordhood information in sentences. The semantic information of bigram plays a similar role to the lexicon, while it is unnecessary to build a large lexicon for ancient Chinese corpus. To deal with the ambiguity, or uncertainty in sentences, we use MC-dropout method to find uncertain parts of sentences. Next we use a Knowledge Fusion Model to retrieve auxillary knowledge and re-predict the uncentain parts. Our experiments show that our model outperforms pre-trained BERT model `https://huggingface.co/SIKU-BERT/sikuroberta` with CRF and Jiayan `https://github.com/jiaeyan/Jiayan` in our dataset *Zuozhuan*.

2. Background and Related Work

2.1. CWS and POS Tagging

Chinese Word Segmentation (CWS) is the fundamental of Chinese natural language understanding. It splits a sentence into several words, which are basic components of a Chinese sentence. CWS is necessary because there is no natural segmentation between Chinese words. Part-of-Speech Tagging (POS Tagging) further assigns POS tags for each word in a sentence.

2.2. Knowledge Retrieval

Knowledge retrieval is a method used to enhance the performance of language models, and they are most commonly used in NER tasks. Knowledge databases (Qiu et al., 2014; Gu et al., 2018) and search engines (Geng et al., 2022) are used to retrieve knowledge, and the knowledge retrieved is used to argument the input sentences.

3. Approach

As previous work (Qiu et al., 2019; Ke et al., 2020), the CWS and POS Tagging task is viewed as a character-based sequence labeling problem. Specifically, given input sequence $X = [c_1, c_2, ..., c_n]$ composed of continuous characters, the model should output a label sequence $Y = [y_1, y_2, ...y_n]$ with $y_i \in TagSet$.

In this section, we will introduce the improvement proposed for local semantic information capture, followed by the uncertainty sampling method. Finally, we will introduce our overall framework utilizing the uncertainty sampling method.

3.1. Local Semantic Enhancement

BERT (Devlin et al., 2018) is a Transformer based bidirectional language model, which solves the problem of long-term dependence in RNN models. However, this also makes BERT lose the ability to capture local semantic features. Therefore, we integrated the bigram features to introduce local semantic information. The overall architecture of our baseline model is displayed in Figure 1, and we call it *Semantic Enhancement BERT*.

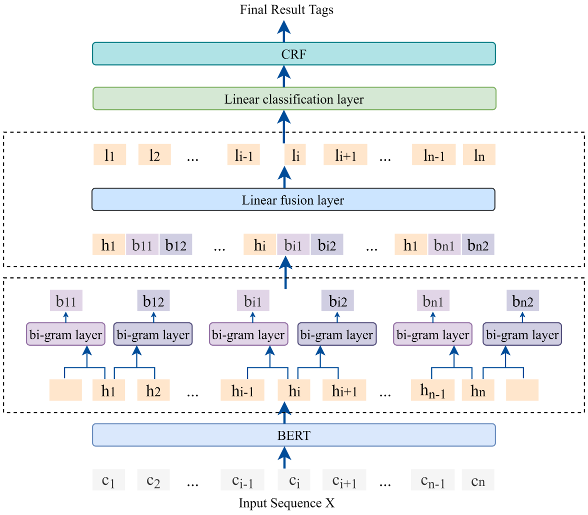
Final Result Tags
CRF
Linear classification layer
l1 l2 ... li-1 li li+1 ... ln-1 ln
Linear fusion layer
h1 b11 b12 ... hi bi1 bi2 ... h1 bn1 bn2
b11 b12 bi1 bi2 bn1 bn2
bi-gram layer
h1 h2 ... hi-1 hi hi+1 ... hn-1 hn
BERT
c1 c2 ... ci-1 ci ci+1 ... cn-1 cn
Input Sequence X

Figure 1: Architecture of baseline model.

3.1.1. Encoder

Given input sequence $X = [c_1, c_2, ..., c_n], X \in \mathbb{R}^n$. We employ BERT as our basic encoder, converting X to hidden character representations as follows,

$$H = BERT(X), \tag{1}$$

where $H \in \mathbb{R}^{n \times d_h}$.

3.1.2. Linear Bigram Layer

The vocabulary of ancient Chinese is short, consise and meaningful, and the bigram features have proved beneficial for CWS (Chen et al., 2017; Ke et al., 2020). Therefore, we construct the bigram concatenated vectors for every character c_i by concatenating it's hidden character representations with the previous character's and the latter character's. Then we convert the concatenated vectors to bigram feature vectors b_{i1}, b_{i2} by two Linear bigram layer as follows,

$$b_{i1} = LinearLayer_1(h_{i-1} \oplus h_i), \tag{2}$$

$$b_{i2} = LinearLayer_2(h_i \oplus h_{i+1}), \tag{3}$$

where $b_{i1}, b_{i2} \in \mathbb{R}^{d_b}$.

3.1.3. Linear Fusion Layer

We construct Composite feature vector h_i for character c_i by concatenating h_i, b_{i1} and b_{i2} as follows,

$$h_i^{'} = h_i \oplus b_{i1} \oplus b_{i2}, \tag{4}$$

where $h_i^{'} \in \mathbb{R}^{(d_h + 2 \times d_b)}$.
$H^{'}$ is defined as follows,

$$H^{'} = [h_1^{'}, h_2^{'}, ..., h_n^{'}]. \tag{5}$$

Then, we use a simple fusion mechanism to convert the Composite feature vectors to Fusion feature vectors by a Linear Layer,

$$L = LinearLayer_3(H^{'}), \tag{6}$$

where $H^{'} \in \mathbb{R}^{n \times (d_h + 2 \times d_b)}, L \in \mathbb{R}^{n \times d_l}$.

3.1.4. Decoder

The Fusion feature representations are converted into the probabilities over the POS labels by an MLP layer,

$$P^T = Softmax(WL^T + b), \tag{7}$$

where $P \in \mathbb{R}^{n \times d_t}$. d_t is the number of POS tags. P_{ik} represents the probability that the label of c_i is tag_k.
Finally, we decode P using **Viterbi algorithm** to obtain the final tag sequence $T = [t_1, t_2, ..., t_n], T \in \mathbb{R}^n$.

3.2. Uncertainty Sampling

BERT is already very powerful. Under the condition that the annotated dataset is very limited, simply increasing the complexity of the model structure will not make performance better. So we introduce uncertainty sampling and knowledge retrieving.

3.2.1. Uncertainty Sampling Method

MC Dropout (Gal and Ghahramani, 2016) is a general approach to obtain the uncertain components. Formally, given input sequence X, we first obtain the provisional label sequence T_p utilizing trained baseline model. Then, we utilize MC dropout to keep dropout active and generate k candidate label sequences $T_1, T_2, ..., T_k$ with Viterbi decoding. The difference between each candidate-predicted word set and the provisional-predicted word set can be considered uncertain words. Then we obtain uncertain components by merging all overlapping uncertain words.

3.2.2. Preliminary Statistics

Similar to Geng et al. (2022)'s evaluation approach, we conduct an investigation on test set of two Ancient Chinese datasets to verify the importance of the uncertainty component. We use *Semantic Enhancement BERT* as baseline model and generate 8 candidate label sequences using MC dropout. The results are displayed in Table 3.

	Zuozhuan	*Shiji*
CWS F1 Score	95.606%	93.465%
CWS Oracle F1 Score	97.777%	96.780%
POS F1 Score	91.229%	87.618%
POS Oracle F1 Score	95.602%	93.417%
$\text{ACC}_{uncertain}$	57.190%	55.951%
$\text{ACC}_{certain}$	94.560%	91.704%

Table 1: The statistics of the uncertain components. **F1 Score** denotes the F1 score of the baseline model on the test dataset. **Oracle F1 Score** denotes the F1 score obtained by the baseline model if the labels of the uncertain components are corrected. $\textbf{ACC}_{uncertain}$ and $\textbf{ACC}_{certain}$ denote the label accuracy of the provisional results for the uncertain components and the confident components, respectively.

The significant gap between certain components and uncertain components indicates that the uncertain components are real hard components and become bottlenecks for performance. Therefore, by querying about uncertain components, the ancient corpus with the same structure can be retrieved.

3.2.3. Retrieving

Different from the retrieval idea in the NER task (Geng et al., 2022), we first collect several Pre-Qin ancient texts to form our knowledge corpus. For word w corresponding to each uncertain component, we query the sentences containing w. In particular, if the uncertain component contains only one character, we construct bigram words w_1 and w_2 for the character w by concatenating it with the previous character and the latter character. Then we look for sentences containing w_1 or w_2 instead of w.

We rank sentences by similarities in order to obtain sentences with grammatical structures similar to X. Generally, the similarity between two sentences P and Q is defined as follows,

$$\text{s} = \frac{\text{union}(\text{P}, \text{Q})}{\|\text{P}\| + \|\text{Q}\|}, \tag{8}$$

where $union(P, Q)$ is the total number of the same characters in P and Q, $\|\text{P}\|$ and $\|\text{Q}\|$ is the length of P and Q, respectively. Finally, we choose the most similar sentences as auxiliary knowledge.

3.3. Framework

In this part, we will present our overall framework, which is displayed in Figure 2.

3.3.1. Stage One: Provisional Results and Uncertainty Sampling

Given input sequence $X = [c_1, c_2, ..., c_n]$, we employ baseline model to obtain the provisional label sequence T_p and candidate label sequences. Then we obtain the uncertain component $U = [c_i, c_{i+1}, ...c_{i+o}]$ using the method in Section 3.2.

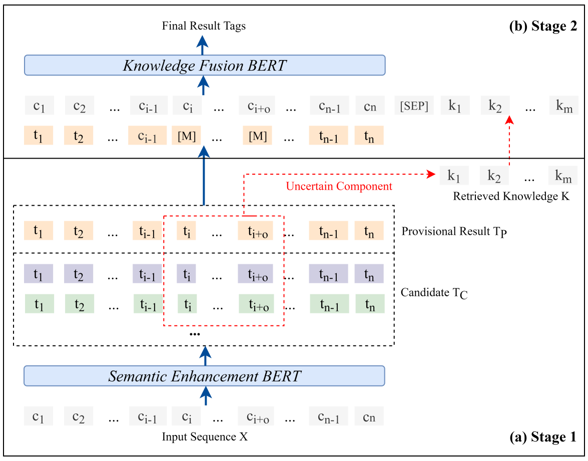
Final Result Tags
(b) Stage 2
Knowledge Fusion BERT
c1 c2 ... ci-1 ci ... ci+o ... cn-1 cn [SEP] k1 k2 ... km
t1 t2 ... ci-1 [M] ... [M] ... tn-1 tn
Uncertain Component
k1 k2 ... km
Retrieved Knowledge K
t1 t2 ... ti-1 ti ... ti+o ... tn-1 tn
Provisional Result Tp
Candidate TC
...
Semantic Enhancement BERT
c1 c2 ... ci-1 ci ... ci+o ... cn-1 cn
Input Sequence X
(a) Stage 1

Figure 2: The overall framework.

If X has no uncertain component, T_p will be taken as the final prediction label sequence T. Otherwise, we use U to retrieve the auxiliary knowledge K. If there are multiple uncertain components, we retrieve them separately and process them independently using the method in Stage Two.

3.3.2. Stage Two: Knowledge Fusion Prediction

In the second stage, we re-predict the label sequence of input sequence X by combining the auxiliary knowledge K and the provisional label sequence T_p obtained in Stage One.

Similar to Geng et al. (2022), we concatenate X and K to obtain the knowledge-enhanced input sequence $X' = [c_1, c_2, ..., c_n, [SEP], k_1, k_2, ..., k_m]$ and construct the auxilary label sequence as follows,

$$t_i' = \begin{cases} t_i & \text{if } i \leq n \text{ and } c_i \notin U \\ [MASK] & \text{if } c_i \in U \\ [PAD] & \text{if } i > n \end{cases}, \tag{9}$$

$$T' = [t_1', t_2', ..., t_n', t_{n+1}', ..., t_{n+m+1}']. \tag{10}$$

Finally, we combine X' and T' as the input of Bert-based *Knowledge Fusion BERT* (KF-BERT) to obtain the probability distribution D,

$$E_{T'} = LabelEmbedding(T'), \tag{11}$$

$$E_{X'} = CharacterEmbedding(X'), \tag{12}$$

$$D = KF\text{-}BERT(E_{T'} + E_{X'}), \tag{13}$$

where $D = [d_1, d_2, ...d_n]$ and d_i is the probability distribution of c_i, and d_{ij} is the probability of c_i being predicted to tag_j.

Label Embedding and *Character Embedding* are parameters need to be trained. Finally, we get the final label sequence by **Viterbi algorithm**. In particular, if there are multiple uncertain components in X, we process them separately in the second stage and average all obtained D before Viterbi decoding.

Model	Test-*Zuozhuan*		Test-*Shiji*	
	CWS-F1(%)	POS-F1(%)	CWS-F1(%)	POS-F1(%)
Jiayan	82.022	/	83.141	/
Siku-RoBERTa + CRF	96.073	91.998	92.937	87.466
SE-BERT	96.018	92.019	93.092	87.594
SE-BERT^{+}	96.148	92.292	93.914	86.691
SE-BERT^{+}+KF-BERT	**96.284**	**92.410**	**93.596**	**87.873**
BERT-Bigram	96.009	91.853	93.015	87.574

Table 2: Jiayan is an NLP toolkit focusing on ancient Chinese processing. SE-BERT denotes *Semantic Enhancement BERT* using Siku-RoBERTa, SE-BERT^{+} denotes *Semantic Enhancement BERT* using Siku-RoBERTa^{+} as pre-trained BERT, and KF-BERT means *Knowledge Fusion BERT* using Siku-RoBERTa. BERT-Bigram denotes Siku-RoBERTa incorporating pre-trained bigram embedding. To utilize the entire training set, we use cross-validation and average the prediction results of K models, where K = 5.

4. Experiment

We conducted a series of experiments to validate the effectiveness of our framework. We follow the competition EvaHan2022 `https://circse.github.io/LT4HALA/2022/EvaHan`, using a tag set containing 22 POS tags and a tag set {B, M, E, S} to denote the beginning, middle, and end of a word as well as single words. Thus we have a total of 88 tags for joint CWS and POS Tagging classification. We used the standard F1-Score as evaluation metric. All experiments were conducted on a server with 8 GeForce RTX 3090.

4.1. Overall Performance

Table 2 shows the over all performance and some ablation experiments.

From Table 2, the performance of our model is much higher than the ancient Chinese processing toolkit Jiayan. our efforts in both semantic enhancement (sikuroBERTa+CRF and SE-BERT) and knowledge fusion (SE-BERT^{+} and SE-BERT^{+}+KF-BERT) show that large improvements were achieved. Also, furtherpretrain of BERT on relevant domain datasets can further improve the performance (as seen for SE-BERT^{+}). Our final model combines all the advantages and achieves good results.

5. Discussion

Regarding the combination of bigram features, we did not introduce new knowledge or more complex structures in our framework. Ke et al. (2020) incorporated pre-trained bigram embedding into their model. Referring to the work of Ke et al. (2020), we conducted another experiment.

The experiment result in Table 2 shows that *Semantic Enhancement BERT* works better than $BERT\text{-}Bigram$. However, the idea still shows a good direction for future research. The ancient vocabulary is short and rich in meaning, and the performance may be further improved if well pre-trained N-gram embedding can be properly introduced.

6. Conclusion

In this paper, we propose a framework for ancient Chinese CWS and POS Tagging that implements semantic enhancement and knowledge fusion. By utilizing bigram features and re-predicting the uncertain samples by fusing knowledge, our framework makes good predictions.

7. References

Chen, X., Shi, Z., Qiu, X., and Huang, X. (2017). Adversarial multi-criteria learning for chinese word segmentation. *arXiv preprint arXiv:1704.07556*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Geng, Z., Yan, H., Yin, Z., An, C., and Qiu, X. (2022). Turner: The uncertainty-based retrieval framework for chinese ner. *arXiv preprint arXiv:2202.09022*.

Gu, J., Wang, Y., Cho, K., and Li, V. O. (2018). Search engine guided neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Ke, Z., Shi, L., Meng, E., Wang, B., Qiu, X., and Huang, X. (2020). Unified multi-criteria chinese word segmentation with bert. *arXiv preprint arXiv:2004.05808*.

Li, X., Yan, H., Qiu, X., and Huang, X. (2020). Flat: Chinese ner using flat-lattice transformer. *arXiv preprint arXiv:2004.11795*.

Qiu, X., Huang, C., and Huang, X.-J. (2014). Automatic corpus expansion for chinese word segmentation by exploiting the redundancy of web information. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1154–1164.

Qiu, X., Pei, H., Yan, H., and Huang, X. (2019). Multi-criteria chinese word segmentation with transformer. *arXiv preprint arXiv:1906.12035*.

Yang, J., Zhang, Y., and Liang, S. (2018). Subword encoding in lattice lstm for chinese word segmentation. *arXiv preprint arXiv:1810.12594*.

Appendix: Datasets and Hyperparameters

The training and test datasets for this experiment are from the competition EvaHan2022 `https://circse.github.io/LT4HALA/2022/EvaHan`. The training and testa were excerpted from *Zuozhuan* and the testb was excerpted from the *Shiji*. The statistical information of the datasets is shown in Table 3

	Size	Length$_{avg}$
Train-*Zuozhuan*	1083K	22.415
Test-*Zuozhuan*	185K	20.902
Test-*Shiji*	352K	29.302

Table 3: Dataset statistics.

	SE-Bert	KF-Bert
Epochs	20	20
Batch Size	32	32
Weight Decay	0.1	0.1
Dropout	0.1	0.1
Learning Rate	1e-5	1e-5
Optimizer	AdamW	AdamW
Warm Up Ratio	0.1	0.1
Max Seq$_{L}en$	128	128
α	-	{0.1,1}

Table 4: Hyper parameters for *Semantic Enhancement Bert* and *Knowledge Fusion BERT*.

The hyper parameters are listed in table 4.

To enhance the learning of uncertain component, we introduce weight coefficient ω_i to set different weights for uncertain components and certain components so that the model pays more attention to the prediction of uncertain parts. The loss function L is defined as Eq. (14),

$$L = \frac{\sum_{\mathrm{i}}^{1\leq \mathrm{i} \leq \mathrm{n}} \omega_{\mathrm{i}} \cdot \mathrm{loss}_{\mathrm{i}}}{\sum_{\mathrm{i}}^{1\leq \mathrm{i} \leq \mathrm{n}} \omega_{\mathrm{i}}}, \tag{14}$$

$$\omega_{\mathrm{i}} = \begin{cases} 1 & \text{if } c_i \in U \\ \alpha & \text{if } c_i \notin U \end{cases}, \tag{15}$$

where ω_i is the weight coefficient at position i. $loss_i$ is the cross-entropy loss at position i. α is a hyper parameter ranges in $[0, 1]$. In particular, we do not make predictions for auxiliary knowledge, nor do we calculate the loss of this part.