|  |
| --- |
| **EvaHan 2025: Named Entity Recognition in Ancient Chinese with Hybrid Segmentation** |
|  |
| **Zihong Yang, Hengda Zhu** |
|  |
|  |
|  |
|  |

Abstract

This paper presents our approach to Named Entity Recognition (NER) in ancient Chinese texts as part of the EvaHan 2025 evaluation. Our system is based on the pretrained **GujiRoBERTa\_jian\_fan** model and incorporates a **hybrid segmentation strategy** combining **punctuation-based and length-constrained** text splitting. We conduct extensive evaluations on three datasets: **Shiji, Twenty-Four Histories, and Traditional Chinese Medicine (TCM)** texts. Our model achieved **an F1 score of 80.53%**, with the highest performance on **historical texts** but notable challenges in **TCM texts**. Error analysis reveals that **ambiguous entity boundaries and domain-specific terminology** significantly impact recognition accuracy. Future work will explore **domain-adapted pretraining** and **hierarchical tokenization** to enhance performance.

Introduction

Named Entity Recognition (NER) in **ancient Chinese texts** presents unique challenges due to the **absence of standardized segmentation**, the **ambiguity of entity boundaries**, and the **domain-specific terminology** found in historical and medical texts. Previous research has applied **modern pre-trained models such as BERT** to historical language processing tasks, demonstrating promising results [1][2]. However, these models often struggle with **low-resource entity types and rare character sequences**, necessitating the development of specialized approaches.

To address these challenges, **Wang et al. (2023)** introduced the **GujiBERT and GujiGPT** model series, which systematically evaluate the effectiveness of large-scale pretraining on historical text NER tasks. Their work benchmarks performance across multiple datasets using comprehensive metrics, including **accuracy, recall, and F1 score** [3]. This research underscores the importance of domain-adapted pretraining for NER in ancient texts, which remains a key challenge in this field.

In this competition, we leverage the **GujiRoBERTa\_jian\_fan** model within a **closed evaluation setting**, where teams are restricted to using only the provided training data and pretrained models. The datasets for this task are drawn from:

• **Dataset A**: Extracted from the *Shiji*, containing six entity categories.

• **Dataset B**: Sourced from the *Twenty-Four Histories*, with three entity categories.

• **Dataset C**: Derived from *Traditional Chinese Medicine Classics*, comprising six entity categories.

This standardized setup ensures a fair comparison of methodologies and allows for systematic evaluation of NER performance across distinct historical and medical domains.

NER System

The NER system consists of three main components: **data processing, model training, and model testing**.

Data Processing

The raw text files are processed line by line, with empty lines removed. We employ a **hybrid text segmentation strategy** that integrates:

• **Maximum length constraints**: Ensures each segment remains within the allowed input size.

• **Punctuation-based segmentation**: Splits sentences at meaningful boundaries.

• **Stop-word-based segmentation**: Enhances coherence by considering common stop words.

After segmentation, **labels are converted into numerical representations**, and the text is tokenized using the **GujiRoBERTa\_jian\_fan** tokenizer to generate **text\_id** and **attention\_mask**. To maintain uniformity, **text\_id, attention\_mask, and labels are padded** to a fixed length.

To streamline the subsequent training process, we implemented a **DataProcess class** that inherits from the **Dataset class** in PyTorch, overriding the \_\_getitem\_\_ method. This allows for efficient batch retrieval of input\_ids, attention\_mask, and labels.

Model Training

The **preprocessed training data** is first converted into batches using **PyTorch’s DataLoader**. The training process involves:

1. **Model Training Loop**: Iteratively updates model parameters over multiple epochs.

2. **Loss Computation**: Uses **CrossEntropyLoss** to calculate classification errors.

3. **Optimization**: Employs **AdamW** optimizer to adjust weights.

The key hyperparameters used during training are summarized in **Table 1**.

|  |  |
| --- | --- |
| **Attribute Name** | **Value** |
| Optimizer | AdamW |
| Loss Function | CrossEntropyLoss |
| Learning Rate | 5e-5 |
| Epochs | 5 |
| Batch Size | 64 |

**Table 1: Training Attributes**

Once training is complete, the **trained model and tokenizer are saved** for future inference.



Model Testing

The testing process follows these steps:

1. **Preprocessing**:

• Tokenization is performed similarly to training data.

• Unlike training, [CLS] and [SEP] tokens are **removed** after tokenization.

• The text is segmented using **punctuation-based or length-constrained strategies**.

2. **Prediction**:

• The processed test data is **batched using DataLoader**.

• The trained model generates **predicted labels**.

3. **Post-processing**:

• Predictions are **filtered** to exclude padding (attention\_mask[i] = 1).

• Final results are **saved in a .txt file** for evaluation.

Results

The performance of our model in the **Close Modality** is summarized in **Table 2**. The evaluation metrics, **Precision (P), Recall (R), and F1 score (F1),** are reported for each of the three datasets (**Test A, Test B, and Test C**), as well as the overall performance across all datasets.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Set** | **Precision (P)** | **Recall (R)** | **F1 Score (F1)** |
| **Test A** | 84.42 | 73.86 | 78.79 |
| **Test B** | 86.65 | 85.71 | 86.18 |
| **Test C** | 70.33 | 83.09 | 76.18 |
| **Total** | 79.69 | 81.40 | 80.53 |

**Table 2: Model Performance on Different Datasets**

From the table, we can observe the following:

• The model performed **best on Test B**, with a precision of **86.65%**, recall of **85.71%**, and F1 score of **86.18%**. This suggests that the model was able to identify entities in **Test B** with high accuracy and completeness.

• For **Test A**, the precision was **84.42%**, while the recall was lower at **73.86%**, leading to an F1 score of **78.79%**. The lower recall may indicate that certain entities were **missed**, potentially due to the complexity or variety of entities in this dataset.

• **Test C** showed a **balanced performance**, with a precision of **70.33%** and a recall of **83.09%**, leading to an F1 score of **76.18%**. The relatively lower precision suggests some **false positives**, where entities may have been incorrectly identified.

• **Overall**, the model’s performance across all datasets achieved an **F1 score of 80.53%**, indicating a solid overall ability to recognize named entities in **ancient Chinese texts**.

The performance comparison between our model and the baseline is further illustrated in **Figure 1** below.

A graph with lines and numbers

AI-generated content may be incorrect.

**Figure 1: Performance Comparison Between Our Model and Baseline**

The figure highlights how our model compares with the baseline in terms of **Precision, Recall, and F1 Score** across the three test sets. While our model achieves slightly higher F1 scores overall, **Test C remains challenging**, showing lower precision and recall compared to the other datasets.



Discussion

In this section, we analyze the results, compare our model with the baseline, discuss the challenges encountered during model training, and propose potential improvements.

**Results Analysis**

The model performed well on **Test B**, suggesting that the entities in this dataset are more consistent or easier to identify. However, the model’s performance on **Test A** and **Test C** was notably lower, especially in terms of **precision**. This highlights the discrepancy between **precision and recall**, indicating that while the model is good at identifying entities when it makes predictions, it struggles to capture all relevant entities.

Notably, our model’s **overall performance was lower than the baseline in all test cases, particularly for Test C**. This suggests that while our model is competitive, it still has **significant room for improvement**, particularly in recall for complex domain-specific texts.

**Comparison with the Baseline**

Compared to the baseline model, our system shows **weaker performance across all datasets**, especially in precision and recall for **Test C (Traditional Chinese Medicine texts)**. The baseline model might have been better at handling certain types of entities due to **better entity-specific tokenization** or a **more optimized training procedure**.

This comparison provides a valuable benchmark and indicates that our model’s **architecture or training strategy may require refinement**. The gap between our model and the baseline suggests several potential areas for improvement, including **more effective segmentation techniques** and **domain-adapted pretraining**.

**Challenges in Model Performance**

• **Entity Boundary Ambiguity**: Ancient Chinese texts often lack clear entity delimiters, making segmentation-based NER difficult.

• **Domain-Specific Vocabulary**: Medical texts contain highly specialized terms that are not well represented in standard pre-trained embeddings.

• **Imbalanced Entity Categories**: Certain entity types (e.g., **medicinal formulas, symptoms**) have far fewer examples than others, affecting model generalization.

**Model Improvements**

To improve performance, future work should focus on:

• **Domain-Specific Pretraining**: Fine-tuning **GujiRoBERTa** on domain-specific corpora (e.g., **medical literature, historical records**) to enhance recognition accuracy.

• **Hierarchical Tokenization**: Adopting **lattice-based models** or **multi-level tokenization** to improve handling of **ambiguous entity boundaries**.

• **Knowledge Augmentation**: Incorporating **domain knowledge graphs** or **lexicon-based constraints** to reduce false positives.

**Future Comparisons**

Although our model is competitive compared to previous approaches in **ancient language processing**, the performance gap with the baseline highlights the need for further **development and refinement**. Future evaluations comparing our system with **other state-of-the-art models** will provide deeper insights into the **strengths and weaknesses** of our approach, guiding further improvements.

References

1. Lample, G., Ballesteros, M., Subramanian, S., et al. (2016). “Neural Architectures for Named Entity Recognition.” Proceedings of NAACL-HLT 2016. Available at: https://www.aclweb.org/anthology/N16-1030.pdf

2. Devlin, J., Chang, M. W., Lee, K., et al. (2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” Proceedings of NAACL-HLT 2019. Available at: https://arxiv.org/abs/1810.04805

3. Wang, D., Liu, C., Zhao, Z., Shen, S., Liu, L., Li, B., Hu, H., Wu, M., Lin, L., Zhao, X., & Wang, X. (2023). GujiBERT and GujiGPT: Construction of Intelligent Information Processing Foundation Language Models for Ancient Texts. Available at:https://arxiv.org/abs/2307.05354