Multi-Modal Multi-Granularity Tokenizer for Chu Bamboo Slips

Yingfa Chen[1*], Chenlong Hu[1*], Cong Feng[2*†], Chenyang Song[1], Shi Yu[1], Xu Han[1], Zhiyuan Liu[1†], Maosong Sun[1]
[1]Tsinghua University
[2]Henan Polytechnic University
chenyingfa1999@gmail.com

Abstract

This study presents a multi-modal multi-granularity tokenizer specifically designed for analyzing ancient Chinese scripts, focusing on the Chu bamboo slip (CBS) script used during the Spring and Autumn and Warring States period (771-256 BCE) in Ancient China. Considering the complex hierarchical structure of ancient Chinese scripts, where a single character may be a combination of multiple sub-characters, our tokenizer first adopts character detection to locate character boundaries. Then it conducts character recognition at both the character and sub-character levels. Moreover, to support the academic community, we assembled the first large-scale dataset of CBSs with over 100K annotated character image scans. On the part-of-speech tagging task built on our dataset, using our tokenizer gives a 5.5% relative improvement in F1-score compared to mainstream sub-word tokenizers. Our work not only aids in further investigations of the specific script but also has the potential to advance research on other forms of ancient Chinese scripts.[1]

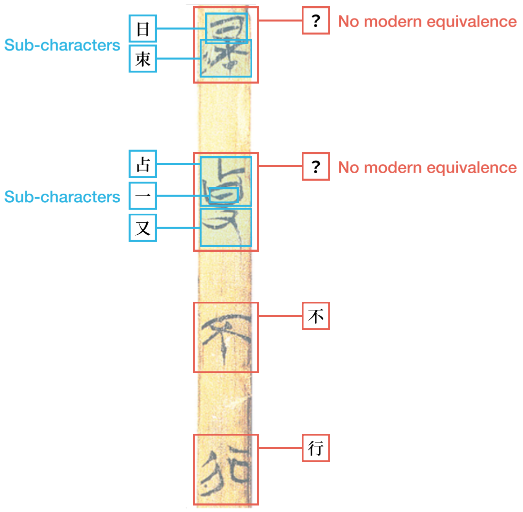


Figure 1: Overview of our proposed tokenizer on an example. Each ancient character is mapped to a modern character if possible. Otherwise, the tokenizer rolls back to decomposing the character into sub-character units, potentially containing useful information. One possible deciphering of the text is "At first, action is not simple". The slip shown is the 14th slip in Zhonggong document from the Shanghai Museum Slips.

1 Introduction

Deep neural networks have demonstrated remarkable success in various natural language processing tasks (OpenAI, 2023; Touvron et al., 2023) as well as the analyses of ancient languages (Sommerschield et al., 2023). Inspired by these former works, we aim to apply deep learning to the analysis of ancient Chinese scripts. However, this application faces three challenges: (1) Most of these ancient scripts are stored as images, which are more difficult to analyze than texts. (2) A large proportion of the characters is rare or undeciphered, making it challenging to train data-driven neural networks. This also implies that the widely-used subword tokenizers such as BPE (Sennrich et al., 2016) and SentencePiece (Kudo and Richardson, 2018) fall short because the neural networks struggle to learn informative representations of the rare and undeciphered characters. (3) Current tokenizers struggle to generalize to unseen materials, in which there is a considerable ratio of out-of-vocabulary (OOV) characters.

To overcome these challenges, we propose a novel multi-modal multi-granularity tokenizer tailored for ancient Chinese scripts, focusing on the 2000-year-old Chu bamboo slip (CBS) script from ancient China. The tokenization pipeline begins by detecting and ordering the characters in image scans of the raw materials into a sequence of char-

[*] Equal contributions

[†] Corresponding authors

[1] The code and data can be found at https://www.github.com/THUNLP/Chujian

Proceedings of the 31st International Conference on Computational Linguistics, pages 6201–6211
January 19–24, 2025. ©2025 Association for Computational Linguistics

acter images. Next, each character is recognized within a pre-defined vocabulary. If the recognition confidence is low, the tokenizer rolls back to tokenizing the character into *sub-character components* (components that make up Chinese characters and are larger than a stroke, and smaller than a character) which may contain rich information about the semantics or phonetics of the text (Sun et al., 2014; Nguyen et al., 2017; Si et al., 2023a).

To demonstrate the effectiveness of our tokenizer, we collect and release the first dataset of CBS texts. It contains 102,722 annotated CBS character images, from 5,033 slips and 164 documents. To facilitate further investigation, we have developed a user-friendly platform where researchers with different expertise can access and analyze the dataset with ease. The proposed tokenizer significantly outperforms the existing baselines, especially on the task of part-of-speech tagging.

The main contributions of this study can be summarized as follows:

1. We collect, process, and release CHUBS, the first large-scale dataset on Chu Bamboo Slip script in a format that is convenient for typical NLP workflows.

2. We propose an annotation scheme for providing useful information about the sub-character features of CBS scripts to address the large proportion of out-of-vocabulary characters prevalent in CBS.

3. Based on the sub-character annotations, we propose a multi-granularity tokenizer that outperforms ordinary character-based tokenizers on downstream tasks.

4. We build a platform for easy access to the data for researchers of all backgrounds to facilitate future research.

2 Related Works

Tokenization Tokenization is the process of splitting a sentence into units. It is essential to current natural language processing techniques and have an integral impact on downstream performance (Mielke et al., 2021). Current NLP tokenizers accept text sequences as inputs and split them into pieces that are then turned into integers to be handled by neural networks. In this work, although the tokenization process start from the image scan of text inscriptions, the goal is to convert the raw representation into a sequence of simple representations that are easy for the pipeline to handle. Therefore, we call our method a tokenization pipeline.

Chinese Tokenization Regarding Chinese characters, most existing tokenization methods operate on the character level. Each token is either once character or a combination of character (Si et al., 2023a). Such method disregard the fact that each Chinese character is composed of components that encode information that may be useful for analyzing the language. Numerous works have shown that tokenizing characters at the sub-character level can improve the downstream performance of Chinese, Japanese, and Korean neural models. Some notable works include Sun et al. (2014); Li et al. (2015); Song et al. (2018); Si et al. (2023a), which have shown that utilizing sub-character components can improve the quality of learned embeddings, as measured by improved performance or efficiency in a wide range of language understanding tasks compared to conventional tokenizers. For pre-trained language models, Si et al. (2023a) show that converting Chinese characters to sub-character sequences can improve the efficiency and robustness in general language understanding. In language generation, Wang et al. (2022) have showed that using stroke information can improve English-Chinese translations.

Deep Learning Applications in Ancient Scripts As a result of the recent advances in the capabilities of deep neural networks in computer vision and natural language processing, there have been numerous works that utilize deep learning methods to assist research in ancient scripts (Sommerschield et al., 2023). Some examples include ancient Greek (Assael et al., 2022), Devanagari (Narang et al., 2021), ancient Chinese (Zhang and Liu, 2021), ancient Japanese (Clanuwat et al., 2019), etc. To the best of our knowledge, our work represent the first attempt to apply deep learning methods in the processing of Chu bamboo slips.

3 Dataset

We begin with a brief introduction to the background of the CBSs (Section 3.1). Then, we describe the collection process of our dataset called CHUBS (**CHU B**amboo **S**lips) (Section 3.2). Finally, we present an open platform for convenient access to our data, especially for researchers of different backgrounds (Section 3.3).

3.1 Chu Bamboo Slips

CBSs are the writing materials used in ancient China during the Warring States period over two thousand years ago, and the earliest known large-scale form of calligraphic writing[2]. The study of it holds great linguistic, historical, and cultural value, especially for East Asian scripts. The content includes, for instance, the oldest known records of ancient classics such as the *Book of Documents* (also the *Classic of History*, Chinese: 尚书) and *Classic of Poetry* (Shijing, Chinese: 诗经).

The slips survived over two thousand years mainly because they were submerged in water until excavation, protecting them from oxidation. For the same reason, most current slips are found along the Yangtze River. As shown in Figure 2, the form of the slips is highly regular, most are 45cm long and 0.6cm wide. The longer slips typically carry between 27 to 38 characters. Multiple slips are tied together to form documents. A real example of a CBS is given in Appendix A.

3.2 CHUBS

Digitizing and understanding CBSs, especially in the view of natural language processing, are of great value in promoting history, culture, and art research studies. However, to the best of our knowledge, there is no public large-scale collection of CBS data prepared in an accessible format that is convenient for usage in typical workflows within the machine learning community. Thus, to facilitate the application of machine learning to aid research in CBS, we collect and publish the first dataset of CBS inscriptions, called CHUBS. It includes high-quality scanned images of the slips and their text annotations.

3.2.1 Data Source

All data is extracted and processed from publicly released textbooks or records by paleographers, containing image scans and transcriptions of a set of bamboo slips from certain excavation projects. We supplemented the materials with some missing transcriptions and extracted the images of the characters from the slip images.

These materials are widely known in the community of paleographers in ancient Chinese scripts. Our contribution is that we are the first to compile these materials into an easily accessible format for the application of machine learning methods. Compiling and analyzing such data is a common practice within the paleography field in ancient Chinese script, and we have consulted with legal experts to make sure that this data can be released under a permissive license such as Apache 2.0.

Since all image scans are extracted and processed from publicly released textbooks containing unearthed materials from various sources and different periods, variations between scans produced by different teams are inevitable. For example, some scans are black, while others are in color.

Table 1 lists each of the data sources as well as the number of documents, slips, and characters from each source. It is worth noting that many of the sources do not have an official English name. Therefore, we only give the pinyin transcription of the Chinese name. We suggest interested readers use the Chinese name when possible for future research.

3.2.2 Annotating Sub-Character Components

Each character is annotated with modern Chinese text. However, manual inspection reveals that at least 27% of the characters in our dataset are not within the set of modern Chinese words[3] (these characters do not have a UTF-8 encoding). In other words, 27% of the detected characters are out of vocabulary (OOV) if we tokenize them on character-level granularity. The upper two characters in Figure 1 are examples of such OOV characters.

There are two reasons for this high proportion of OOV CBS characters:

1. The CBS character has not yet been deciphered due to drastic changes in character forms or material degradation.

2. The CBS character does not have a modern Chinese equivalent (but experts believe that they know the meaning of the character).

These CBS characters are annotated with a set of *sub-character components* including, but not limited to radicals or *pianpangs*[4]. For example, assuming that the character “想” (pronunciation: *xiang*,

[2] Some Oracle Bone Script were formed by brushes, but only in extremely small amounts.

[3] A word may consist of multiple characters.

[4] Radicals are components of Chinese characters traditionally used for indexing in dictionaries. Each character has exactly one radical. *Pianpangs* is a superset of radicals. For more details, please refer to https://en.wikipedia.org/wiki/Chinese_character_internal_structures

Source name	Chinese name	# documents	# slips	# characters
Tsinghua University Slips	清华简	50	1,402	31,468
Shanghai Museum Slips	上博简	60	881	25,795
Baoshan Slips	包山简	4	337	12,647
Guodian Slips	郭店简	18	705	11,865
Geling Slips	葛陵简	8	743	6,209
Zenghouyi Slips	曾侯乙简	4	198	6,016
Jiudian Slips	九店简	2	232	2,956
Wangshan Slips	望山简	3	273	2,218
Changtaiguan Slips	长台关简	3	148	1,504
Zidanku Silk	子弹库帛	7	7	1,471
Yangtianhu Slips	仰天湖简	1	42	335
Wulipai Slips	五里牌简	1	18	109
Xiyangpo Slips	夕阳坡简	1	2	54
Ynagjiawan Slips	杨家湾简	1	38	41
Caojiagang Slips	曹家岗简	1	7	34
Total		164	5,033	102,722

Table 1: The amount of data from different sources of our collection of CBSs.

meaning *think*) does not have a modern equivalence, it may be labeled as “相心”[5] (pronunciation: *xiang xin*). If even such sub-character components are unrecognizable, it is annotated with a placeholder to indicate that the character is unrecognizable.

However, there is no common consensus on how to split Chinese characters into sub-character components. Our approach is based on the philosophy that each unit should be semantically or phonetically meaningful (i.e., it is a morpheme or a phoneme). This is because we hypothesize that further splitting such units does not provide additional useful information about the text but may introduce noise or result in unnecessarily lengthy token sequences.

Concretely, we request an expert in the field (with a Ph.D. degree studying CBS) to annotate each CBS character with the corresponding sub-character components. One possibility is to label the pianpang. However, this has two main limitations when applied to CBS scripts. Firstly, CBS characters are very different from modern Chinese; not every CBS character has a pianpang. Secondly, we want to retain as much information about the character as possible, so we need a method for annotating the semantics or phonetics of the part of the characters that is not the pianpang.

3.2.3 Sub-Character Component Annotation Scheme

Addressing the above limitations, our final annotation procedure is as follows. For a given CBS character, if it is already labeled with a modern Chinese character (i.e., it is not OOV), we keep it as it is. Otherwise, we first identify it as one of the three types of Chinese characters: **logograms** (*xiangxing* characters, Chinese: 象形字), **semantic-phonetic compound characters** (*xingsheng* characters, Chinese: 形声字), and **phonograms** (*jiajie* characters, Chinese: 假借字). Such classification of Chinese characters was first introduced by Chen (1956), and is commonly taught in Chinese schools[6]. Then, we start with a sub-character vocabulary with 540 items introduced by *Shuowen Jiezi* (Xu, 1963), a well-known Chinese dictionary released around 100 CE during the Eastern Han dynasty.

- For **semantic-phonetic compound characters**, we split them into the semantic and phonetic parts (the former is always a logogram), and apply the following rules.

- For **logograms** and **phonograms**: we try to split it into components of the current sub-

[5] We have refrained from using a more advanced encoding system (such as including the positioning of the components) to keep the annotation cost low.

[6] This categorization scheme is called “three category theory” (*san shu shuo*, Chinese: 三书说), but there are also other categorization methods. Two notable instances are “four category theory” and “six category theory”.

character vocabulary. If there exists a part of the character that is not and does not include any of the current sub-character components, we add that part as a sub-character component into the vocabulary.

Repeating this process for all characters in our library results in 798 sub-character components in total, which makes up our final sub-character vocabulary.

We emphasize that the vocabulary construction may have considerable impact on the downstream performance, but it is out of the scope of this thesis work.

3.3 Open Platform

To better foster future research in CBS scripts, we build and release a platform to make accessing our data more convenient for researchers from different backgrounds. The platform allows the download of the entire collection as well as searching particular images based on the text annotation, origin, and character appearance (searching by hand-written strokes), which is essential for searching for characters without modern Chinese equivalents. Further, this platform also features pipeline processing capabilities for CBS, including detecting, recognizing, and retrieving characters, significantly reducing both time and human resources for experts. Specifically, for a CBS image, it can detect each character and recognize it with our multi-modal tokenizer. Appendix B displays a screenshot of this platform.

4 Multi-Modal Multi-Granularity Tokenizer

In summary, our tokenizer consists of multiple neural networks that perform object detection and classification in a pipeline. The input is the image of the material containing the Ancient inscriptions. The pipeline consists of the following steps:

1. The characters in the bamboo slip are detected using an object detection model, cropped then ordered into a sequence based on their location.

2. Each image is fed to a character recognition that maps the CBS characters into a modern Chinese character/word.

3. If the classification confidence is lower than a certain threshold, the tokenizer falls back to sub-character analysis by recognizing the sub-character components of the character.

The output is a sequence where each element is either a single character or a set of sub-character components. The classification confidence threshold is typically determined using a validation set of examples from the downstream task.

4.1 Sub-Character Recognition

As mentioned in Section 3.2.2, many characters in our dataset are not within the set of modern Chinese words. For such characters, assigning a unique class would not be conducive, because the class label may not help us better understand the ancient text. Therefore, we propose to recognize the sub-character components[7] of the characters instead. This may be beneficial for downstream tasks because Chinese character components may represent rich information about the phonetics and semantics of the character.

This is done with a multi-label classifier whose vocabulary is simply the set of 798 sub-character components we have annotated in CHUBS.

5 Experimental Details

5.1 Models

Character Detection Specifically, we employ the YOLOv5 model (Jocher et al., 2020), one of the most used versions in the YOLO series (Redmon et al., 2016). We train this model on the CBS images annotated by domain experts. We highlight that the annotation process is rather simple. It involves indicating the bounding box of each CBS character.

Character and Sub-Character Recognition For both character and sub-character recognition, we try both ResNet (He et al., 2016) and Visual Transformer (ViT) (Dosovitskiy et al., 2020), which are two strong models with great capabilities in image classification. We use roughly the same number of parameters for both architectures. The difference between character and sub-character recognition is the number of classes and that the former is an ordinary multi-class classification while the latter is a multi-label classification.

Specifically, we start from commonly used public checkpoints, the official `resnet152` model of PyTorch and the ViT by Wu et al. (2020)[8]. These model checkpoints are pre-trained on ImageNet

[7] We use "components" to refer to any consistent and frequent set of strokes smaller than or equal to a character.

[8] https://huggingface.co/google/vit-base-patch16-224

(Deng et al., 2009), and we finetune them on CHUBS.

5.2 Training Data

Detector Training Data To train the CBS character detector, an expert paleographer is asked to manually annotate a small number of CBS. The annotations are then quality-checked by other authors. In total, 177 image scans of bamboo slips from Tsinghua University Slips were annotated, of which 141 were used as training data, and 36 for validation. We emphasize that this annotation process is rather simple because most CBS characters are very easy to identify in the image scans.

Classifier Training Data The character and sub-character recognizer are simply trained on CHUBS, since the data already contains all supervision needed. The frequency distribution of the characters follows a Zipfian distribution, so approximately half of the characters only appear once in the dataset. To ensure that each class contains enough data for both training and testing, we discard characters that have less than k images (we use $k = 3, 10$ in character recognition and $k = 2, 20$ in sub-character recognition). We then split the data into training, validation, and test sets by an 8:1:1 ratio, while ensuring that the test set has at least one example from every class.

5.3 Training Details

All training experiments are conducted on an A100 GPU, and implemented with PyTorch. We use the Adam optimizer (Kingma et al., 2020) and a learning rate scheduler that decays by 0.9 after every epoch. We only search different batch sizes and maximum learning rates during the hyperparameter search to keep the computational cost low.

6 Results

Since the tokenization pipeline has three steps, we first show the empirical performance of each part. Then, we apply the tokenizer on an example downstream task, part-of-speech (POS) tagging, to demonstrate its effectiveness over character-based tokenizers (one CBS character per token).

6.1 Character Detection

The performance of the character detector is shown in Table 2. The *near-perfect* F1-score implies that the model is well-suited and robust for CBS characters and that it introduces minimal noise to our tokenization pipeline. This is in line with the intuition that determining the boundaries of CBS characters is rather simple for human annotators. Based on these detection results, we then conduct character recognition.

Precision	**Recall**	**F1**
0.998	0.996	0.997

Table 2: Character Detection Results with YOLOv5.

6.2 Character Recognition

The result of the character recognizer on the test set is shown in Table 3, in which we can see that ViT consistently outperforms ResNet, which is consistent with the results by the authors of ViT. The high accuracy for $k = 10$ indicates that the model provides great practical value in analyzing CBS characters. Meanwhile, we see considerably lower accuracies for $k = 3$, which implies that analyzing infrequent characters is still a challenging research question.

6.3 Sub-Character Recognition

Table 4 shows the performance of the sub-character recognition module. Perhaps surprisingly, ResNet beats ViT by a large margin, which differs from the observation in the character recognition experiments. One possible explanation for this is that each head in the multi-head attention module is responsible for recognizing a certain set of components (or their corresponding features), but the number of classes is too great for the architecture. Further investigations are outside this work's scope.

6.4 Downstream Task: Part-of-Speech Tagging

To demonstrate the effectiveness of our multi-granularity tokenizer, we apply it to a part-of-speech (POS) tagging task in the CBS script.

We create a POS tagging dataset for CBS by manually annotating 1,109 randomly sampled sentences using the BIO (Beginning, Inside, and Outside) format (Ramshaw and Marcus, 1999). This annotation is conducted by an expert in CBS scripts. Then, we apply our multi-granularity tokenizer and a character-based tokenizer (each character is one token).

Our annotations include the following ten parts-of-speech that are commonly found and analyzed in ancient Chinese:

Model	Top-1	Top-3	Top-5	Top-10
		$k = 3$		
ResNet	61.23	65.48	70.84	72.33
ViT	73.48	84.65	87.45	89.95
		$k = 10$		
ResNet	72.60	83.70	87.18	90.57
ViT	90.11	95.03	96.06	97.16

Table 3: Accuracy (in %) of character recognition models on the test set. k indicates the minimum occurrence of a character in the dataset.

Method	Recall	Precision	F1
		$k = 2$	
ResNet	84.79	77.32	80.88
ViT	22.48	26.31	24.24
		$k = 20$	
ResNet	85.70	78.31	80.19
ViT	28.57	28.23	28.40

Table 4: Recognition result (in %) of sub-character components of our model.

1. Noun (Chinese: 名词, *mingci*)
2. Verb (Chinese: 动词, *dongci*)
3. Conjunction (Chinese: 连词, *lianci*)
4. Adjective (Chinese: 形容词, *xingrongci*)
5. Adverb (Chinese: 副词, *fuci*)
6. Numeral (Chinese: 数量词, *shuliangci*)
7. Modal Particle (Chinese: 语气词, *yuqici*)
8. Pronoun (Chinese: 代词, *daici*)
9. Preposition (Chinese: 介词, *jieci*)
10. Auxiliary Word (Chinese: 助词, *zhuci*)

This dataset will be publicly released along side with our CHUBS dataset and training code.

Tokenizer	Recall	Prec.	F1
Character-based	47.9	43.8	45.3
Multi-granularity	50.2	46.1	47.8

Table 5: The part-of-speech performance (in %) when using a conventional character-level tokenizer (Char-Tokenizer) and our multi-granularity tokenizer.

When splitting characters into sub-character components, the label corresponding to the components is the same as the label for the original character. Then, a special token representing the boundary between each character is added to the sides of the sequence of components for each character. The predictions for these special tokens are ignored.

For the downstream model, we tune a large language model for this task using in-context learning. Specifically, we randomly sample 10 examples from the training data to use as in-context demonstrations and prompt the LLM to generate the predicted entities and the types as a Markdown list. The actual prompt template will be given along with the code after the review period. We use GPT-3-Turbo with default hyperparameters and repeat the experiments with 10 random seeds to ensure reproducibility.

6.4.1 Part-of-Speech Tagging Results

The result is shown in Table 5. We observe that using our multi-granularity tokenizer can significantly (with a p-value of 0.0079 in a t-test) improve the POS tagging performance of the downstream model, as we have expected.

6.4.2 Error Analysis

To gain more insight, we manually inspect examples in which the multi-granularity tokenizer predicts correctly while the character-based tokenizer does not. We find that the majority of such erroneous predictions involve characters that the character recognizer fails to recognize.

In CBS, it is common that one character can be written in different forms, but these forms usually share a common part. One representative example is “我捷灭夏”[9] (*wo jie mie xia*, meaning “I/we quickly overthrew the Xia dynasty”). In this slip, the second character is rare in the dataset and the last two characters are written in the atypical forms that included extra parts in addition to the most typical form. Conseqeutnly, the character-level recognizer can only recognize the first character, hence, the POS tagger can only predict the first character’s POS correctly. The POS tagger based

[9] This is one sentence from the “Yin Hao” (Chinese: 尹浩) document from the Tsinghua University Slips.

on the multi-granularity tokenizer can correctly predict the POS of the other characters because it successfully recognizes the parts that resemble the typical form of the characters, which is enough to comprehend the semantics of the sentence.

7 Conclusion and Discussions

We have proposed a multi-modal multi-granularity tokenizer for better analyzing ancient Chinese scripts than the existing more popular sub-word tokenizers. We have also collected the first large-scale multi-modal dataset of CBS text with an open platform targeted at audiences of different backgrounds. We believe this work is an important step in leveraging deep learning methods in the research of East Asian scripts.

What are the differences between multi-granularity tokenizers and mainstream tokenizers? Currently, most tokenizers are a kind of "subword tokenizer". This includes Byte-Pair Encoding (BPE) (Sennrich et al., 2016) and SentencePiece (Kudo and Richardson, 2018), used in GPT-4 (OpenAI, 2023) and LLaMa (Touvron et al., 2023), respectively. The tokens in these tokenizers are often called *sub-words* (sequences of characters that are smaller than space-delimited words but larger than letters). For Chinese, mainstream tokenizers usually treat each Chinese character as an atomic unit. In contrast, our multi-granularity tokenizer splits each Chinese character into smaller sub-character components and provides this information to the downstream neural network.

Why is identifying sub-character components conducive to downstream tasks? Tokenizers that treat each character as an atomic unit generally work for phonetic languages such as Indo-European languages because splitting phonetic letters into smaller components typically provides little to no additional information[10]. However, for ideographic languages such as Chinese, components of a character may encode rich information about the semantics or phonetics of the characters. For common characters, the model may be able to learn such information automatically from that distribution of co-occurring characters. However, for infrequent characters or unknown characters, the sample size of co-occurring characters is too small. Although some works have shown that language models can implicitly learn the letter composition (which is a kind of sub-token information) of tokens (Si et al., 2023b; Hiraoka and Okazaki, 2024), it is reasonable to hypothesize that such information requires large amounts of training data and tokenization at the sub-character level can provide conductive bias that either enhances performance or reduces the amount of data required to achieve the same performance.

Limitations

In terms of the performance of the tokenizer, there are many possible methods for improving the effectiveness of the components of our tokenizer, such as pre-training on a corpus of modern text, larger/better model architectures, and better data pre- or post-processing. Moreover, augmenting the tokenizer with more knowledge about the history may help. But, we have not employed more tricks to keep our analysis simple.

Also, although we have demonstrated the effectiveness of our tokenizer on the CBS script, it may be less effective on other scripts due to variations between scripts. However, since many other scripts face the same challenges highlighted in the introduction, our method should still have a performance advantage over conventional tokenizers. Additionally, due to the annotation cost, we have only investigated our tokenizer's effectiveness on one downstream task.

Ethical Concerns

This work presents a new dataset on the Chu bamboo slips, a writing material from ancient China over two thousand years ago. We also introduce a new tokenizer for better processing ancient Chinese scripts with a large number of characters that do not have modern Chinese correspondence. The goal is to advance research in this ancient script as well as other forms of ancient Chinese scripts, which should not have significant ethical implications. However, the original content from these raw materials may have ethical implications for certain groups, but since these are existing historical materials, we do not make efforts to censor any content.

Acknowledgements

This work is supported by the National Key R&D Program of China (No. 2022ZD0116312), Na-

[10] One possible example task that we hypothesize might benefit from splitting Latin characters into multiple tokens is for answering questions about the shape of the letters.

tional Natural Science Foundation of China (No. 62236004), and Institute Guo Qiang at Tsinghua University.

References

Yannis Assael, Thea Sommerschield, Brendan Shillingford, Mahyar Bordbar, John Pavlopoulos, Marita Chatzipanagiotou, Ion Androutsopoulos, Jonathan Prag, and Nando de Freitas. 2022. Restoring and attributing ancient texts using deep neural networks. *Nature*, 603(7900):280–283.

Mengjia Chen. 1956. *Yin Xu Pu Ci Survey*.

Tarin Clanuwat, Alex Lamb, and Asanobu Kitamoto. 2019. Kuronet: pre-modern japanese kuzushiji character recognition with deep learning. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 607–614. IEEE.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Tatsuya Hiraoka and Naoaki Okazaki. 2024. Knowledge of pretrained language models on surface information of tokens. *Preprint*, arXiv:2402.09808.

Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, ChristopherSTAN, Liu Changyu, Laughing, tkianai, Adam Hogan, lorenzomammana, yxNONG, AlexWang1900, Laurentiu Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Francisco Ingham, Frederik, Guilhen, Hatovix, Jake Poznanski, Jiacong Fang, Lijun Yu, changyu98, Mingyu Wang, Naman Gupta, Osama Akhtar, PetrDvoracek, and Prashant Rai. 2020. ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements.

Diederik P Kingma, J Adam Ba, and J Adam. 2020. A method for stochastic optimization. arxiv 2014. *arXiv preprint arXiv:1412.6980*, 106.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Conference on Empirical Methods in Natural Language Processing*.

Yanran Li, Wenjie Li, Fei Sun, and Sujian Li. 2015. Component-enhanced chinese character embeddings. *arXiv preprint arXiv:1508.06669*.

Sabrina J. Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y. Lee, Benoît Sagot, and Samson Tan. 2021. Between words and characters: A brief history of open-vocabulary modeling and tokenization in nlp. *ArXiv*, abs/2112.10508.

Sonika Rani Narang, Munish Kumar, and Manish Kumar Jindal. 2021. Deepnetdevanagari: a deep learning model for devanagari ancient character recognition. *Multimedia Tools and Applications*, 80:20671–20686.

Nguyen, Julian Brooke, and Timothy Baldwin. 2017. Sub-character neural language modelling in japanese. In *SWCN@EMNLP*.

OpenAI. 2023. Gpt-4 technical report.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Chenglei Si, Zhengyan Zhang, Yingfa Chen, Fanchao Qi, Xiaozhi Wang, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2023a. Sub-character tokenization for Chinese pretrained language models. *Transactions of the Association for Computational Linguistics*, 11:469–487.

Chenglei Si, Zhengyan Zhang, Yingfa Chen, Fanchao Qi, Xiaozhi Wang, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2023b. Sub-character tokenization for Chinese pretrained language models. *Transactions of the Association for Computational Linguistics*, 11:469–487.

Thea Sommerschield, Yannis Assael, John Pavlopoulos, Vanessa Stefanak, Andrew Senior, Chris Dyer, John Bodel, Jonathan Prag, Ion Androutsopoulos, and Nando de Freitas. 2023. Machine learning for ancient languages: A survey. *Computational Linguistics*, pages 1–44.

Yan Song, Shuming Shi, and Jing Li. 2018. Joint learning embeddings for chinese words and their components via ladder structured networks. In *International Joint Conference on Artificial Intelligence*.

Yaming Sun, Lei Lin, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2014. Radical-enhanced chinese character embedding. In *Neural Information Processing: 21st International Conference, ICONIP 2014, Kuching, Malaysia, November 3-6, 2014. Proceedings, Part II 21*, pages 279–286. Springer.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Zhijun Wang, Xuebo Liu, and Min Zhang. 2022. Breaking the representation bottleneck of chinese characters: Neural machine translation with stroke sequence modeling. *ArXiv*, abs/2211.12781.

Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. 2020. Visual transformers: Token-based image representation and processing for computer vision. *Preprint*, arXiv:2006.03677.

Shen Xu. 1963. *Shuowen jiezi*. Zhonghua Book Company. In Chinese.

Cheng Zhang and Xingjun Liu. 2021. Feature extraction of ancient chinese characters based on deep convolution neural network and big data analysis. *Computational Intelligence and Neuroscience*, 2021.

A An example of a CBS Material

To better understand the nature of CBS, we give an example of a CBS material in our dataset in Figure 2. The text is read from top to bottom and right to left. Each strip can be viewed as the equivalence of one "line" in a modern document. The three strips together form a document. CBS has no punctuation marks (similar to other forms of Ancient Chinese), and the reader is supposed to infer the pauses and coherence between characters from the semantics.

B Open Platform

The platform described in Section 3.3 will be launched after the anonymous review process. A screenshot of it is shown in Figure 3. The platform is a website, and the interaction system was implemented using the Gradio library.

C AI-Assistant-Related Statement

AI-assisted tools were used for error-checking in writing this paper, and for code-completion during the implementation of the experiments.

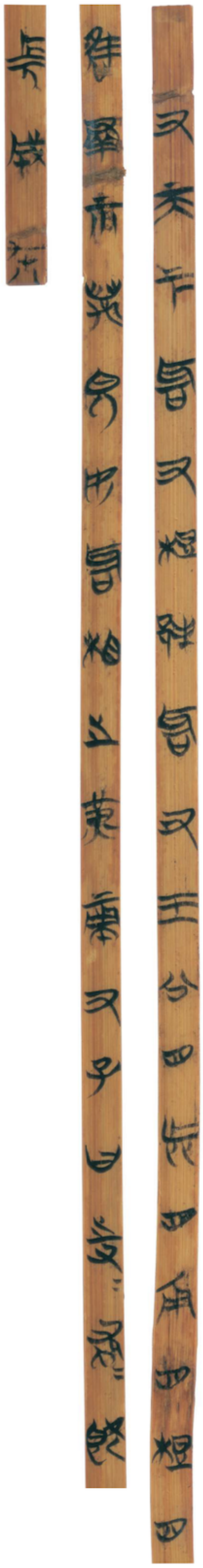
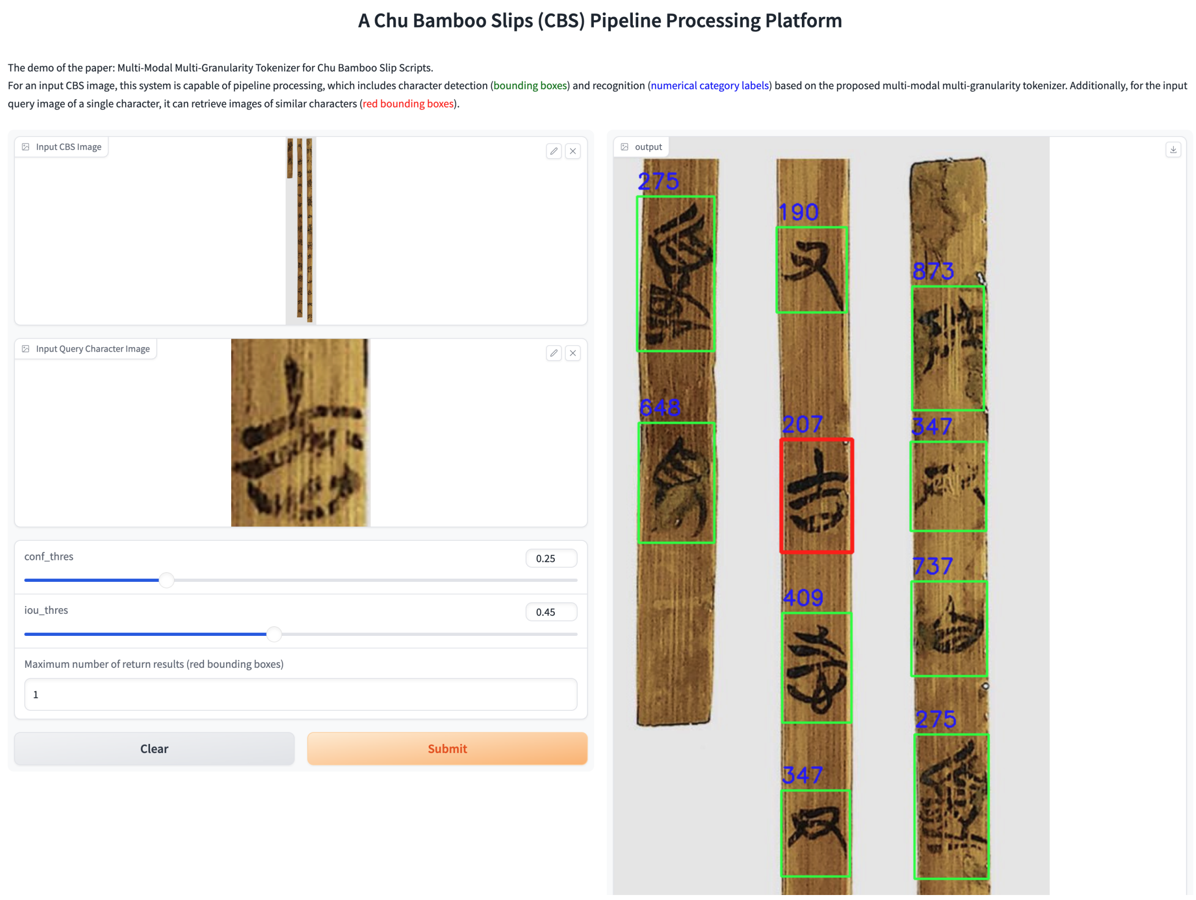


Figure 2: An example of a CBS material. The slip shown is the 98th slip of the "Wu Ji" from Tsinghua University Slips.



A Chu Bamboo Slips (CBS) Pipeline Processing Platform

The demo of the paper: Multi-Modal Multi-Granularity Tokenizer for Chu Bamboo Slip Scripts.

For an input CBS image, this system is capable of pipeline processing, which includes character detection (bounding boxes) and recognition (numerical category labels) based on the proposed multi-modal multi-granularity tokenizer. Additionally, for the input query image of a single character, it can retrieve images of similar characters (red bounding boxes).

conf_thres 0.25

iou_thres 0.45

Maximum number of return results (red bounding boxes)

1

Figure 3: A screenshot of our platform for accessing the dataset and a demo of our tokenizer.