Translation via Annotation: A Computational Study of Translating Classical Chinese into Japanese

Zilong Li
Department of Linguistics
University of Colorado, Boulder
Zilong.Li@colorado.edu

Jie Cao
School of Computer Science
University of Oklahoma
jie.cao@ou.edu

Abstract

Ancient people translated classical Chinese into Japanese using a system of annotations placed around characters. We abstract this process as sequence tagging tasks and fit them into modern language technologies. The research on this annotation and translation system faces a low-resource problem. We alleviate this problem by introducing an LLM-based annotation pipeline and constructing a new dataset from digitized open-source translation data. We show that in the low-resource setting, introducing auxiliary Chinese NLP tasks enhances the training of sequence tagging tasks. We also evaluate the performance of Large Language Models (LLMs) on this task. While they achieve high scores on direct machine translation, our method could serve as a supplement to LLMs to improve the quality of character's annotation.[1]

1 Introduction

Classical Chinese (5th century B.C.E. to 19th century A.D.) has a long history in Japan. Although the exact timing of classical Chinese's introduction to Japan remains unclear, the presence of classical Chinese in Japan dates back to at least the 8th century A.D. The two oldest classical Japanese books, *Kojiki* and *Nihon Shoki*, are written entirely in classical Chinese. In Japan, classical Chinese is referred to as *Kanbun*. During the adoption of *Kanbun*, Japanese people developed an annotation and reading system, called *Kundoku*, by which they translated classical Chinese into Japanese. Today, this annotation system continues to play a role in education and humanities research. Understanding this annotation system helps us to build educational software to assist Japanese education. It can also benefit the digital humanities research of East Asian history and literature.

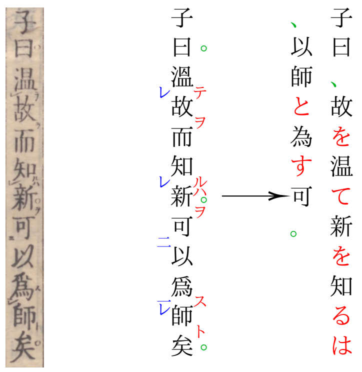
子曰。温故而知新可以爲師矣
子曰、故を温て新を知るは、以師と為す可。

Figure 1: Classical Chinese sentence with marks and its Japanese translation. Green punctuations are **Kutōten** to segment sentences. Blue symbols are **Kaeriten** indicating the reading order. Red characters are **Okurigana** for grammatical and inflectional roles.

In the *Kundoku* translation system, annotations are placed around each Chinese character using three types of marks: **Kutōten**, **Kaeriten**, and **Okurigana**, as illustrated in Figure 1. These marks respectively indicate sentence punctuation, reading order, and grammatical/inflectional information, guiding readers to convert Classical Chinese text into coherent Japanese sentences.

After annotation, people reconstruct Chinese characters with their marks. Chinese characters are first reordered according to the **Kaeriten** marks. Then, **Okurigana** marks are appended to the specific Chinese characters. Through this procedure, a Japanese translation of the given classical Chinese sentence is derived.

In the era of machine learning and Large Language Models (LLMs), researchers are increasingly applying modern language technologies to automate the annotation (Yasuoka, 2020) and translation (Wang et al., 2023b) of *Kanbun*. However, there are two main challenges in developing automatic annotation and translation models for the *Kundoku* system: (1) the lack of parallel corpus annotated with *Kundoku* marks, and (2) the un-

[1] Our code and data are available at https://github.com/shiryusann/KanbunKundoku.

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6031–6045
March 24-29, 2026 ©2026 Association for Computational Linguistics

derexplored impact of classical Chinese linguistic knowledge on language modeling.

In this paper, we present a brief and systematic study of the *Kundoku* annotation and translation system, focusing on addressing these two challenges. We focus exclusively on **Kaeriten** and **Okurigana**, since **Kutōten** (punctuations) are typically already provided in most classical Chinese books. Our contributions are as follows:

- We theoretically analyze the expressiveness of **Kaeriten** marks and design an automaton with transducer that decodes characters with marks into Japanese sentences. This validates our hypothesis that reducing classical Chinese–Japanese translation to *Kundoku* marks tagging tasks is both theoretically sound and practically feasible.
- To alleviate the challenge of the low-resource, we construct a new dataset from online digitized classical Chinese texts and their corresponding Japanese translations. We propose a LLM-based mark generation method, utilizing the automaton for validation to generate *Kundoku* marks.
- To incorporate necessary classical Chinese knowledge, we fine-tune classical Chinese Language models on our new dataset using multi-task supervision with a set of auxiliary Chinese NLP tasks. Ablation studies on these auxiliary tasks provide an understanding of the role of classical Chinese knowledge in this task. Our optimal model outperforms previous baseline across many evaluation metrics, and even performs as comparable to some LLMs.
- We also conduct a comprehensive evaluation of several large language models (LLMs) on this task using both zero-shot and few-shot prompting, shedding light on future opportunities and challenges in applying LLM to this *Kundoku* annotation and translation system.

2 Related Works

The task of translating classical Chinese into Japanese via language technologies was first proposed by Yasuoka (2018). They designed a rule-based method leveraging Universal Dependencies (UD) parsing for classical Chinese, which generates **Kaeriten** marks based on part of speech (POS) tags and the direction and label of dependency arcs. Additionally, they defined specific rules to adjust the position of generated **Kaeriten** marks according to their context. Yasuoka (2020) subsequently introduced a dictionary used to add **Okurigana** marks to Chinese characters. These two components together constitute a complete annotation system as described in the previous section.

Wang et al. (2023b) built a pipeline to directly translate classical Chinese poems into Japanese using pretrained Language Models (PLMs). They decomposed the translation task into two stages: character ordering followed by text generation. In their pipeline, Chinese characters are first processed by BERT/RoBERTa models to determine their order within the target Japanese sentence. Then, the reordered characters are fed into mT5/mGPT models to generate the final translation. In addition to this pipeline, they released their dataset, comprising approximately 3,400 sentence pairs consisting of classical Chinese poems and the corresponding Japanese translations. However, it lacks annotations for **Kaeriten** and **Okurigana** marks.

Beyond research involving pretrained language models, other studies have explored the *Kundoku* annotation system from the perspective of combinatorics and algorithms. Shimano (2009) investigated the expressiveness of **Kaeriten** marks. They proposed a tree-structure model and a matrix model to depict the reading process of Chinese characters. They further proposed a recursive formula that computes the number of character permutations that can be described by **Kaeriten**. In subsequent work, Shimano (2012) solved this recursive formula and derived the corresponding generating function. They summarized their models and demonstrated that the *Kundoku* process is equivalent to a context-free language (CFL) with the Chomsky hierarchy (Shimano, 2018).

In the domain of low-resource Japanese translation research, Mao et al. (2020) incorporated Japanese syntactic knowledge into language models by introducing a word reordering training objective. Mao et al. (2022) extended this strategy to more language pairs and achieved improvements in machine translation.

The translation of classical Chinese has also received attention in recent years. Existing research discusses different aspects of this field, including dataset construction (Wong et al., 2024; Liu et al., 2025), evaluation (Zhou et al., 2023; Bennett et al., 2025; Chen et al., 2025), knowledge retrieval (Wei et al., 2025), time-aware translation (Chang et al.,

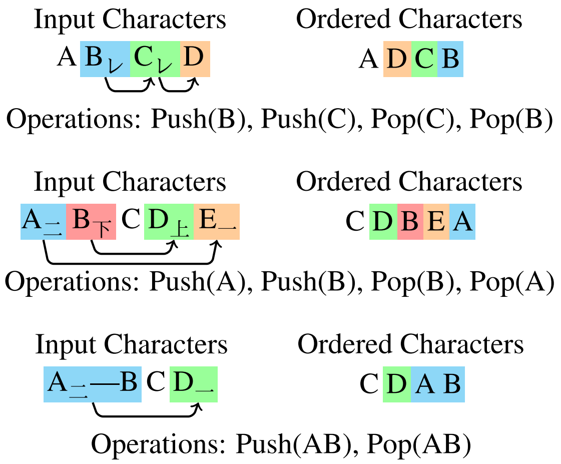

Figure 2: Examples of **Kaeriten**. Sentences on the left are characters with marks. Sentences on the right are characters in the correct order. Black arrows represent characters being read after the target character. Stack operations are listed under each example.

2021), and shared task (Wang et al., 2023a).

3 The Expressiveness of the *Kundoku* Mark System

As introduced in the previous section, **Kaeriten** marks dictate the reading order of Chinese characters in Japanese translation. Since classical Chinese and Japanese have the syntactic divergence, it is crucial to investigate how many permutations of Chinese characters can be expressed by **Kaeriten** marks. Shimano (2012) approached this question from the perspective of combinatorics. In this section, we abstract the reading process into stack operations and address the same problem from a computational perspective. Based on this abstraction, we implement a pushdown automaton (PDA) that decodes Chinese characters with annotations.

3.1 Reading via a Stack Data Structure

Consider a sequence of Chinese characters annotated with **Kaeriten**. For those characters without annotations, their relative position remains unchanged in the final reordered sequence. Therefore, they require no stack operations. For characters with **Kaeriten** marks, their positions are altered. Based on the functions, **Kaeriten** marks can be categorized into three types, as shown below:

レ mark. This mark indicates that the character should appear immediately after its successor in the reordered sequence. As shown in the top example of Figure 2, character B bears a レ mark and is followed by character C. Thus, in the target sequence, B appears immediately after C. In terms of stack operations, we push a character onto the stack, if it holds a レ mark. This character should be popped from the stack, when its following character appears at the end of currently decoded sequence.

一二 like mark. This category encompasses ordinal marks such as 一二, 上下, and 甲乙. This type of marks explicitly dictate the reading order. In the middle example of Figure 2, character A appears immediately after character E, while character B tightly succeeds character D. In stack operations, we push a character annotated with 二, 下 or 乙 onto the stack, when we encounter it. The character is popped from the stack, when its corresponding character with 一, 上 or 甲 has been read and appended to the end of currently decoded sequence. It is worth noting that this type of mark enforces a hierarchy structure. Characters with 上下 must be nested within characters with 一二, as shown in Figure 2. This hierarchy ensures the Last-in First-out (LIFO) principle of stack is preserved during the annotation process.

— mark. This mark functions as a hyphen in English. It serves to connect characters into a single unit. Consequently, other **Kaeriten** marks operate on this unit as a whole instead of an individual character. As shown in the bottom example of Figure 2, characters A and B are moved, pushed, and popped as a single unit.

Except for — mark, the other **Kaeriten** marks operate at the individual character level and are directly related to stack operations. Thus, the number of character sequences that can be expressed by the **Kaeriten** mark system is theoretically equivalent to the number of sequences that can be sorted by a stack. Knuth (1997) discussed the enumeration of such stack sortable permutations. For a sequence of n characters (yielding $n!$ permutations), by Knuth's theorem, the number of sequences that can be expressed by レ mark and 一二 like mark is

$$a_n = \binom{2n}{n} - \binom{2n}{n-1} = \frac{1}{n+1}\binom{2n}{n}$$

When we incorporate — mark, the analysis becomes more complicated, since each element in the stack can represent a sequence of characters. Atkinson and Stitt (2002) referred to this data structure as *a stack of queues*. They derived the generating function of this data structure, which is

$$\frac{1-3x+x^2-\sqrt{1-6x+7x^2-2x^3+x^4}}{2x}$$

Kruchinin and Kruchinin (2013) provided the methodology for deriving the closed-form formula from this generating function. By their theory, the number of sequences among $n!$ permutations of n characters that can be expressed by **Kaeriten** is

$$a_n = \sum_{m=1}^{n+1} \frac{(-1)^{n-m+1}}{m} \binom{m}{n-m+1} \left(\sum_{i=0}^{m-1} \binom{m}{i} \binom{2m-i-2}{m-1} \right)$$

This formula represents the theoretical upper bound of the expressiveness of the **Kaeriten** mark system.[2]

#Chars	#Perms	#Perms by **Kaeriten**	Pct
1	1	1	100
2	2	2	100
3	6	6	100
4	24	20	83.33
5	120	70	58.33
6	720	254	35.28
7	5040	948	18.81
8	40320	3618	9.97

Table 1: Considering n Chinese characters, the number and percentage of permutations of characters that can be expressed by **Kaeriten** marks.

Table 1 describes the number of permutations of n Chinese characters that can be expressed by **Kaeriten** marks. As the sequence length grows, the total number of permutations increases, while the proportion of acceptable permutations diminishes. This phenomenon suggests that natural language syntax imposes strong structural constraints, in spite of large-scale potential character permutations. When people do translation, the constraint by **Kaeriten** marks reduces the reasoning space. The impact of the constraint becomes more pronounced as sentence length increases.

3.2 Automaton and Transducer

Since the reading of レ mark and 一二 like mark can be described as stack operations, we define a Pushdown Automaton (PDA) and transduction operations to read a list of Chinese characters and output them in the target Japanese order. — mark can be easily incorporated into this PDA by regarding characters connected by it as single units in the input alphabet. This PDA reads character-mark pairs and only accepts sentences with valid annotation.

PDA for Kaeriten reading

$L = \{\text{valid } (cm)^n \mid n \geq 1, c \in C, m \in M\}$
where $C = \{\text{Chinese characters}\}$
and $M = \{E, \text{レ}, O_{i,j}, \text{—}\}$ [3]

- **States**: $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- **Input Alphabet**: $\Sigma = C \cup M$
- **Stack Alphabet**: $\Gamma = \Sigma \cup \{Z_0\}$
- **Initial State**: q_0
- **Accepting States**: $F = \{q_4\}$
- **Initial Stack Symbol**: Z_0

Transition Function (δ) and Transduction:

1) $\delta(q_0, c, \sigma) = \{(q_0, c\sigma)\},\ \sigma \neq O_{i,1} \quad \rightarrow \epsilon$
2) $\delta(q_0, c, O_{i,1}) = \{(q_3, O_{i,1})\} \quad \rightarrow \epsilon$
3) $\delta(q_0, \text{レ}, \sigma) = \{(q_0, \text{レ}\sigma)\} \quad \rightarrow \epsilon$
4) $\delta(q_0, E, c\sigma) = \{(q_1, \sigma)\} \quad \rightarrow c$
5) $\delta(q_0, O_{i,j}, \sigma) = \{(q_0, O_{i,j}\sigma)\},\ j > 1 \quad \rightarrow \epsilon$
6) $\delta(q_0, O_{i,1}, \sigma) = \{(q_0, O_{i,1}\sigma), (q_2, O_{i,1}\sigma)\} \quad \rightarrow \epsilon$
7) $\delta(q_1, \epsilon, \text{レ}c\sigma) = \{(q_1, \sigma)\} \quad \rightarrow c$
8) $\delta(q_1, \epsilon, Z_0) = \{(q_0, Z_0)\} \quad \rightarrow \epsilon$
9) $\delta(q_1, \epsilon, \text{レ}O_{i,1}) = \{(q_2, O_{i,1})\} \quad \rightarrow \epsilon$
10) $\delta(q_1, \epsilon, O_{i,j}) = \{(q_0, O_{i,j})\} \quad \rightarrow \epsilon$
11) $\delta(q_2, \epsilon, O_{i,j}cO_{i,j+1}) = \{(q_2, O_{i,j+1})\} \quad \rightarrow c$
12) $\delta(q_2, \epsilon, O_{i,j}cO_{m,n}) = \{(q_0, O_{m,n})\},\ i \neq m \quad \rightarrow c$
13) $\delta(q_2, \epsilon, O_{i,j}cZ_0) = \{(q_0, Z_0)\} \quad \rightarrow c$
14) $\delta(q_2, \epsilon, O_{i,j}c\text{レ}) = \{(q_1, \text{レ})\} \quad \rightarrow c$
15) $\delta(q_0, \epsilon, Z_0) = \{(q_4, Z_0)\} \quad \rightarrow \epsilon$

note: σ represents any element in Γ

The PDA defined above serves as a mechanism to check the validity of **Kaeriten** annotations. The derived transducer facilitates the reordering of Chinese characters. By appending **Okurigana** marks to each reordered character, we obtain the final Japanese translation of the given classical Chinese sentence.

[2] For more information about the generating function and the formula, see https://oeis.org/A078482.

[3] E represents the case that there is no **Kaeriten** mark following the character. $O_{i,j}$ represents 一二 like mark with hierarchy number i and ordinal number j. For instance, 一 is in the first layer of the hierarchy, and it is the first in its group. It is represented as $O_{1,1}$. Similarly, 二 is represented as $O_{1,2}$.

For a better understanding of **Kaeriten** marks and the execution of the PDA, we provide an illustrative example in Appendix C.

4 Dataset Construction

The research on *Kundoku* translation system faces a low-resource challenge. There is only one open source dataset created by Wang et al. (2023b). This dataset contains approximately 3,400 sentences, restricted to the genre of classical Chinese poems from the 7th century A.D. to the 10th century A.D. Their dataset also lacks annotations. This dataset is insufficient for research on the *Kundoku* system.

This section introduces how we constructed a new dataset from the largest online accessible website[4] about classical Chinese and Japanese translations. The data within this website is organized as pairs consisting of a punctuated classical Chinese sentence and its corresponding Japanese translation. However, this dataset still lacks **Okurigana** and **Kaeriten** annotations. Therefore, we propose methods to automatically generate these marks.

4.1 Mark Generation

In the *Kundoku* system, **Okurigana** are Japanese kanas appearing in conjunction with Chinese characters. For most characters, we simply assign the immediately following kanas as their **Okurigana**. However, some Chinese characters are rendered as kanas in Japanese translations. We need to restore these to their original Chinese characters before further processing. Since these characters typically function as interjections or conjunctions in sentences, we identified their positions via part of speech (POS) tagging and constructed a dictionary to map the kanas back to Chinese characters. Since currently available Japanese POS taggers are based on Modern Japanese, to obtain reliable POS tags for the classical Japanese sentences, we first utilized the GiNZA POS tagger (Hiroshi and Masayuki, 2019) to generate raw POS tags. Then, we refined the result using GPT-4o.

In our constructed dataset, we provided a more fine-grained annotation scheme. We distinguished kanas playing grammatical roles from **Okurigana** and classified them as **particle**. Different from **Okurigana**, **particle** is less related with specific Chinese characters and serves to indicate case, tense and so on. This distinction was also achieved through POS tagging analysis. For kanas appearing within the boundary of Japanese words with tag `VERB`, `ADV`, `NOUN`, `ADJ`, `PRON` and `DET`, we regarded them as **Okurigana**.

To generate **Kaeriten** marks, we first aligned each Chinese character in the original classical Chinese sentence with its counterpart in the Japanese translation to determine the reading order. There exists two methods to generate **Kaeriten** marks: building the inverse of the PDA described above, or employing a rule-based approach based on characters' relative positions. As illustrated in Figure 2, we assign ㆑ mark to consecutive characters if they are reversed in the Japanese sentence. We assign ㆒㆓ like mark to non-consecutive characters if they appear together in the Japanese sentence with inverted order. We use — mark to connect characters when they are reordered as a single unit. For the sake of simplicity, we selected the latter method.

4.2 Dataset Statistics and Validation

We collected 9,292 sentences (95,066 characters) of classical Chinese to form the dataset. The dataset covers genres including history, philosophy, military strategy and poetry, across different time periods. Regarding sentence length distribution, 6,099 sentences contain fewer than 10 characters; 2,667 range between 10 and 20; 388 range between 20 and 30; and 138 exceed 30 characters. Our dataset exhibits a better genre coverage and a balanced length distribution, making it highly valuable for the research on the *Kundoku* system.

During the generation of **Kaeriten** marks, we deployed the PDA to check the quality of annotation. Almost all marks generated are accepted by the PDA, except for a few sentences, which contain errors in the original Japanese translations. We manually corrected these sentences and reannotated them. The high acceptance rate demonstrates the validity of our **Kaeriten** marks generation method. This result also underscores the nature of the historical annotation system and the syntactic alignment between classical Chinese and Japanese.

For generated **Okurigana** and **particle** marks, we also manually checked their quality and corrected mistakes. In our dataset, there are 42,473 characters with **Okurigana** or **particle** marks. Among them, we corrected 11,834 characters. The acceptance rate of LLM's annotation is 72.14%. For **Okurigana** and **particle** generation, LLM-based approach is feasible, but still needs human intervention.

[4] `https://kanbun.info/`

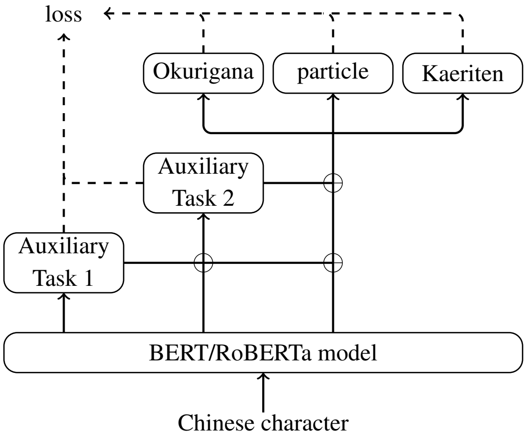

Figure 3: An example of our multitask learning model's structure. Solid lines represent the flow of embeddings and logits. Dashed lines represent the flow of loss.

5 Modeling and Ablation Study

Since *Kundoku* marks are assigned to each individual Chinese character, we formulate the annotation process as sequence tagging tasks and integrate them with modern NLP paradigms. After acquiring all marks, we employ the transducer and construct Japanese translations. There exist BERT/RoBERTa based language models pretrained on classical Chinese, such as sikubert (Li et al., 2022), bert-ancient-chinese (Wang and Ren, 2022) and roberta-classical-chinese-base-char (Yasuoka et al., 2022). These models with deep encoder architecture and character level tokenization are naturally suitable for the sequence labeling tasks.

Linguistic knowledge of Classical Chinese is dispensable for human annotators. Consequently, exploring the impact of injecting extra Chinese knowledge is an interesting research topic.

In this section, we fine-tune several pretrained language models on the dataset constructed above. We adopt a multi-task learning strategy when annotating distinct types of *Kundoku* marks. Furthermore, we conduct an ablation study to quantify the effect of introducing extra linguistic knowledge via auxiliary learning tasks. We evaluate performance of models with machine translation metrics, sequence matching metrics and the PDA pass rate.

5.1 Model Architecture

We adopted a joint learning strategy to predict all mark types simultaneously, and ensured that different classifiers share the same low-level representations so that they may benefit from each other. For auxiliary Chinese NLP tasks, we constructed a multi-step learning architecture (Zhang et al., 2023), analogous to the multi-task learning framework by Hashimoto et al. (2017).

Figure 3 depicts the architecture of an example model with two auxiliary tasks. The output embeddings from the BERT/RoBERTa model are first passed to the classifier of Auxiliary Task 1 to compute the logits. Then, we concatenate the original output embeddings with these logits to form the input for Auxiliary Task 2. Finally, we concatenate the output embeddings with the logits from both Auxiliary Task 1 and Auxiliary Task 2 to form the final input for the three main classifiers. The total model loss is calculated as the weighted sum of the losses derived from all auxiliary tasks and main tasks, as shown below.

$$\begin{aligned} L_{model} = w_a(&L_{\text{aux task 1}} + L_{\text{aux task 2}}) + \\ &w_m(L_{\text{Okurigana}} + L_{\text{particle}} + L_{\text{Kaeriten}}) \end{aligned}$$

5.2 Main Tasks and Auxiliary Tasks

The primary labeling tasks consist of **Okurigana**, **particle** and **Kaeriten** prediction. We adopted marks appearing in our dataset as the label space of these classifiers. In our experiments, **Okurigana**, **particle** and **Kaeriten** have 434, 476 and 20 labels respectively.

In addition to the three main tasks, we adopted four classical Chinese sequence labeling tasks as auxiliary tasks. They are listed below:

Word Segmentation Identifies word boundaries. We use the simplest B-I tag system in experiments.

Part Of Speech Tagging Identifies grammatical roles. We use the Universal Dependencies POS tag system. In experiments, there are 14 labels.

Dependency Arc Labeling Indicates the relative position of a word in the sentence's dependency tree. We utilize the tag system proposed by Gómez-Rodríguez et al. (2023). There are 115 labels in our dataset.

Dependency Type Labeling Mark the type of dependency arc for each word. 44 labels are used in experiments.

All auxiliary labels were generated based on the parse results obtained via HanLP[5](He and Choi, 2021), which offers classical Chinese dependency parsing. We introduced a special `continue` label across POS, dependency arc, and dependency type

[5] https://github.com/hankcs/HanLP

tasks to designate characters that are not the initial of a word. When computing loss, we took `continue` into consideration. Since in classical Chinese, most characters work as a word independently, including `continue` does not hinder training, and even forces models to learn to distinguish word boundaries.

All auxiliary tasks are organized according to the following hierarchy: word segmentation < POS tagging < dependency arc labeling < dependency type labeling. Regardless of the number of auxiliary tasks employed, this relative priority is always maintained.

5.3 Evaluation Metrics

We employ the same metrics used by Wang et al. (2023b) to ensure comparability of experimental results. These metrics are classified into machine translation metrics including **BLEU** (Papineni et al., 2002), **RIBES** (Isozaki et al., 2010), **ROUGE-L** (Lin, 2004), **BERTScore** (Zhang et al., 2020), and character reordering/matching metrics including **Kendall's Tau** (τ) and **Perfect Match Ratio (PMR)**. All metrics are computed at the character level.

In addition to these metrics, we incorporate **chrF (character-level F-score)** (Popović, 2015), **TER (Translation Edit Rate)** (Snover et al., 2006), and the **pass rate of PDA** into the evaluation. The former two metrics assess the quality of Japanese translation, while the latter one evaluates language models' understanding of **Kaeriten** marks.

5.4 Experimental Setting and Result

The generated dataset was randomly shuffled and split into training, validation, and test sets with a ratio of 8:1:1. Experiments were conducted on sikubert, bert-ancient-chinese, roberta-classical-chinese-base-char and bert-base-japanese-char[6]. For each pretrained language model, we trained a classifier to predict *Kundoku* marks with 0, 1, 2, 3, 4 auxiliary classical Chinese NLP tasks incorporated. This resulted in a total of 16 different model configurations for each base language model. To ensure the main tasks remain the primary training objective, we adjusted the loss weights. We assigned main/auxiliary task weights of 8:2, 7:3, and 6:4 for configurations with one, two, and three or more auxiliary tasks, respectively. We adopted the early stopping strategy during training. For more details about the training setting, see Appendix A.

Table 2 presents the evaluation result for roberta-classical-chinese-base-char. Evaluation results for other models are available in Appendix E. Overall, we observe that models based on pretrained classical Chinese models significantly outperform those models pretrained on Japanese. It emphasizes the significance of domain specific language knowledge acquired from pretraining. By comparing performance of models with auxiliary tasks configurations, we observe that Chinese NLP auxiliary tasks generally yield positive effects on the main tasks. Models based on sikubert, roberta-classical-chinese-base-char and bert-base-japanese-char achieve their best performance across many metrics with the inclusion of two auxiliary tasks. However, integrating additional auxiliary tasks leads to a decline in performance. We hypothesize that with more tasks introduced, the learning of main tasks is diluted. The model optimization becomes biased towards auxiliary tasks. Furthermore, with more auxiliary tasks included, the architecture of models gets deeper as well. This increased depth may impede optimization and loss propagation. Finally, the training data of auxiliary tasks was generated by dependency parsers. The parsers may have hallucination and introduce noise into the training data. This may lead to performance degradation.

6 Model Comparison

In this section, we compare our proposed approach against the translation pipeline proposed by Wang et al. (2023b). Their models were evaluated on their classical Chinese poem dataset. To ensure the comparability, we applied identical data processing procedures to their dataset and trained a new model. Based on the ablation study results in the previous section, we used word segmentation and dependency type labeling as auxiliary tasks, since this combination of tasks reached the best machine translation performance. We trained this new model on roberta-classical-chinese-base-char and maintained the same experimental setting.

Table 3 presents the evaluation result of our model alongside the best models built by Wang et al. (2023b). Since they employed distinct models for characters reordering and machine translation, we report the best translation scores from mT5-large and best character ordering scores from

[6] `https://huggingface.co/tohoku-nlp/bert-base-japanese-char`

Tasks	BLEU	chrF	BERTScore	ROUGE-L	RIBES	Kendall's τ	TER	PMR	Pass Rate
Kundoku marks	64.76	59.76	94.57	85.75	59.01	19.87	97.00	94.39	94.09
+seg	62.70	57.10	94.27	84.89	56.31	20.83	97.19	94.50	94.62
+pos	65.41	58.81	94.56	85.48	56.56	19.93	96.88	94.63	**95.38**
+arc	64.99	59.69	94.51	85.52	57.96	19.93	97.27	94.84	92.37
+type	62.33	56.05	94.07	84.38	53.83	21.41	**97.37**	**95.03**	92.37
+seg+pos	65.24	59.71	94.54	85.58	58.01	19.86	96.80	94.51	92.69
+seg+arc	60.44	55.40	93.96	84.05	54.22	21.69	96.84	94.60	94.07
+seg+type	**66.07**	**61.25**	**94.72**	**86.26**	**60.20**	**18.79**	96.95	94.31	93.33
+pos+arc	63.06	58.40	94.22	84.85	57.65	20.57	96.96	94.42	93.23
+pos+type	61.90	58.29	94.22	85.20	59.25	20.62	97.07	94.49	94.84
+arc+type	63.03	59.09	94.31	85.17	58.75	20.32	96.81	94.17	94.41
+seg+arc+type	60.66	56.34	94.07	84.39	56.08	21.41	96.73	94.12	94.19
+seg+pos+type	62.22	57.44	94.26	84.88	56.41	20.71	96.78	94.01	94.19
+seg+pos+arc	61.17	56.17	94.16	84.51	55.06	21.41	96.66	94.02	93.98
+pos+arc+type	62.03	57.00	94.14	84.53	55.65	21.03	96.71	94.34	93.01
+all	62.14	57.09	94.13	84.69	56.9	21.19	96.49	93.93	92.15

Table 2: Experimental result on roberta-classical-chinese-base-char.

roberta-classical-chinese-char. Overall, our approach achieves results comparable to the best models constructed by them. Notably, our model performs better in character ordering and has higher **perfect matching ratio (PMR)**. Consequently, our model achieves superior scores on the translation metrics, such as **RIBES**, which is sensitive to character order. In contrast to the two-stage pipeline, our approach has a smaller number of parameters and also provides character level annotations.

7 How LLMs Perform on These Tasks

Modern large language models (LLMs) have demonstrated their remarkable capabilities in understanding natural languages. It is crucial to assess their performance on our annotation and translation tasks to investigate their strengths and weaknesses of understanding classical Chinese and Japanese. Due to budget constraints, we randomly selected 100 sentences from our test set. We evaluated several most up-to-date models, including DeepSeek-V3.2, Gemini-3-pro-preview, Gemini-3-flash-preview, and GPT-5.2 on these sentences. We employed both zero-shot and few-shot prompting strategies to investigate the impact of in-context learning. First, we instructed LLMs to directly generate Japanese translations of classical Chinese sentences and evaluated the results with machine translation metrics. Subsequently, we prompted these LLMs to assign each Chinese character a **Kaeriten** mark and input their responses to the PDA. We evaluated their annotations on character ordering metrics. To keep consistent on model construction, we used the same base model and auxiliary tasks as used in the previous section.

Table 4 outlines the comparative evaluation results between the LLMs and our approach. All LLMs achieve high scores in machine translation, with few-shot examples significantly enhancing their performance. Gemini models achieve the highest scores on machine translation metrics, while our approach is comparable to some LLMs, such as GPT-5.2 and DeepSeek-V3.2. However, regarding annotation metrics and character ordering, our approach outperforms all LLMs.

The high machine translation scores of LLMs might come from their training data. Since we collected and constructed our dataset from publicly available online resources, these sentences might be included in the training data of LLMs. Few-shot examples likely provide a context that assists LLMs in locating the correct translations. For **Kaeriten** annotation, since there is not enough training data available for LLMs, they have to rely on their reasoning abilities to understand **Kaeriten** marks and character ordering rules. Our model's translation is derived directly from the annotation results. Due to the limited label space, our model's translation output is not as flexible as LLMs. However, in our model, the character ordering and **Kaeriten** annotation benefit from the explicit supervision along with auxiliary Chinese NLP tasks. Consequently, our model achieves higher scores on character ordering metrics.

At the end of experiments, we applied our **Kaeriten** annotation pipeline to Japanese transla-

Models	BLEU	chrF	BERTScore	ROUGE-L	RIBES	TER	Kendall's τ	PMR	Pass Rate
our approach	55.51	52.12	92.58	80.11	63.67	27.13	92.23	90.00	95.41
Wang et al.	51.40	-	93.40	74.70	58.30	-	94.40	78.30	-

Table 3: Evaluation result on the dataset created by Wang et al. (2023b). We trained a **new** model with POS tagging and dependency type labeling as auxiliary tasks and roberta-classical-chinese-base-char as the base model on their dataset.

	Models	BLEU	chrF	BERTScore	ROUGE-L	RIBES	TER	Kendall's τ	PMR	Pass Rate
	our approach	62.50	57.93	94.38	84.96	55.72	21.07	**97.35**	**94.91**	92.00
zero-shot	DeepSeek-V3.2	61.96	57.60	93.73	82.60	53.52	23.46	82.49	62.28	91.00
	Gemini-3-pro-preview	69.09	62.57	95.09	86.45	58.81	20.15	94.96	90.49	97.00
	Gemini-3-flash-preview	68.72	61.88	94.73	85.70	56.58	18.86	94.02	89.72	97.00
	GPT-5.2	59.70	53.38	93.41	80.13	49.41	27.89	75.19	49.86	86.00
few-shot	DeepSeek-V3.2	64.23	57.42	94.34	83.58	53.05	22.33	77.77	58.22	89.00
	Gemini-3-pro-preview	**73.23**	**67.03**	**95.75**	88.13	**61.24**	**15.94**	95.42	91.01	96.00
	Gemini-3-flash-preview	72.79	65.86	95.56	**88.16**	59.74	16.11	95.13	91.21	**98.00**
	GPT-5.2	62.87	55.96	94.17	82.52	49.63	23.34	72.73	48.96	87.00

Table 4: Performance on Randomly selected 50 sentences in our test set. Machine translation metrics scores are evaluated on generated Japanese sentences. Character ordering scores are evaluated on generated **Kaeriten** marks after the processing of PDA. The middle part corresponds to the result of zero-shot experiments, while the lower part shows the result of few-shot experiments.

tions directly generated by the few-shot Gemini-3-pro-preview. We obtained new **Kaeriten** marks and did the same evaluation process. In this setting, the **Kendall's** τ score and the **PMR** score improved to 96.18 and 93.72 respectively. This result indicates that LLMs possess the implicit character ordering knowledge from Chinese to Japanese, but they sometimes do not explicitly express in the annotation. Our annotation method can serve as an effective supplement to LLMs for correctly generating marks.

8 Conclusion

Japanese people translate classical Chinese into Japanese via annotation. In this work, we formulate this annotation process within the modern NLP paradigm as sequence tagging tasks. **Kaeriten** plays a central role in the syntactic reconstruction of Japanese sentence. We demonstrate that the annotation and reading of **Kaeriten** marks can be abstracted as sorting a sequence with a stack. We derive the theoretical upper bound of the expressiveness of **Kaeriten** and construct a pushdown automaton to validate annotation quality and generate reordered characters. To alleviate the low-resource problem, we construct a new dataset with annotations from online open source data. During the construction, we validate the effectiveness of the PDA and highlight the tight syntactic relation between classical Chinese and Japanese. Furthermore, we develop multi-task learning models and observe the benefit of introducing auxiliary Chinese NLP tasks. To achieve the best performance, empirical results suggest that the number of auxiliary tasks should not be more than two. Finally, we evaluate different LLMs on these tasks. Our evaluation of LLMs reveals that LLMs generate very high-quality translations, but there is still room for improvement in annotating **Kaeriten** marks.

Limitations

Despite introducing a substantial dataset, the size of training data remains quite limited. We are still facing a low-resource challenge. We anticipate that by digitizing more books annotated by ancient people, we could address the low-resource problem.

Our dataset is semi-automatically constructed using LLMs. Since LLMs are not well trained on classical languages, they introduce mistakes in the generated annotation. We still need to make an effort to correct generated marks. This process needs well trained annotator and time-consuming.

Additionally, our current classification of *Kundoku* marks is based on linguistic intuition. This classification is coarse and not optimized for practice. Having a fine-grained set of marks might constrain the sampling space more effectively and yield superior results.

Acknowledgments

We appreciate the work of Web Kanbun-Taikei, the biggest online resource available for *Kundoku* research. The contributors collect *Kanbun* documents and provide standard translations. We also thank the ARR reviewers and ACs for their insightful comments for improving the paper.

References

M.D. Atkinson and T. Stitt. 2002. Restricted permutations and the wreath product. *Discrete Mathematics*, 259(1):19–36.

Eric R. Bennett, HyoJung Han, Xinchen Yang, Andrew Schonebaum, and Marine Carpuat. 2025. Evaluating evaluation metrics for Ancient Chinese to English machine translation. In *Proceedings of the Second Workshop on Ancient Language Processing*, pages 71–76, The Albuquerque Convention Center, Laguna. Association for Computational Linguistics.

Ernie Chang, Yow-Ting Shiue, Hui-Syuan Yeh, and Vera Demberg. 2021. Time-aware Ancient Chinese text translation and inference. In *Proceedings of the 2nd International Workshop on Computational Approaches to Historical Language Change 2021*, pages 1–6, Online. Association for Computational Linguistics.

Andong Chen, Lianzhang Lou, Kehai Chen, Xuefeng Bai, Yang Xiang, Muyun Yang, Tiejun Zhao, and Min Zhang. 2025. Benchmarking LLMs for translating classical Chinese poetry: Evaluating adequacy, fluency, and elegance. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33007–33024, Suzhou, China. Association for Computational Linguistics.

Carlos Gómez-Rodríguez, Diego Roca, and David Vilares. 2023. 4 and 7-bit labeling for projective and non-projective dependency trees. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6375–6384, Singapore. Association for Computational Linguistics.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple NLP tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1923–1933, Copenhagen, Denmark. Association for Computational Linguistics.

Han He and Jinho D. Choi. 2021. The stem cell hypothesis: Dilemma behind multi-task learning with transformer encoders. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5555–5577, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mai Hiroshi and Masayuki. 2019. 短単位品詞の用法曖昧性解決と依存関係ラベリングの同時学習. 言語処理学会第25回年次大会.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, Cambridge, MA. Association for Computational Linguistics.

D.E. Knuth. 1997. *The Art of Computer Programming: Fundamental Algorithms, Volume 1*. Pearson Education.

Vladimir V. Kruchinin and Dmitry V. Kruchinin. 2013. Composita and its properties. *Preprint*, arXiv:1103.2582.

Bin Li, Yiguo Yuan, Jingya Lu, Minxuan Feng, Chao Xu, Weiguang Qu, and Dongbo Wang. 2022. The first international Ancient Chinese word segmentation and POS tagging bakeoff: Overview of the EvaHan 2022 evaluation campaign. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 135–140, Marseille, France. European Language Resources Association.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu, Lan Lan, Jiahuan Cao, Hiuyi Cheng, Kai Ding, and Lianwen Jin. 2025. Large-scale corpus construction and retrieval-augmented generation for Ancient Chinese poetry: New method and data insights. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 779–817, Albuquerque, New Mexico. Association for Computational Linguistics.

Zhuoyuan Mao, Chenhui Chu, and Sadao Kurohashi. 2022. Linguistically driven multi-task pre-training for low-resource neural machine translation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 21(4).

Zhuoyuan Mao, Fabien Cromieres, Raj Dabre, Haiyue Song, and Sadao Kurohashi. 2020. JASS: Japanese-specific sequence to sequence pre-training for neural machine translation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3683–3691, Marseille, France. European Language Resources Association.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Tatsuo Shimano. 2009. 漢文訓読の数学モデル—返り点と竪点が生み出す漢字列の総数—. In 計量国語学会第五十三回大会予稿集. 計量国語学会.

Tatsuo Shimano. 2012. 漢文訓読における返り点・竪点システムの数学的構造について. In 計量国語学会第五十六回大会予稿集. 計量国語学会.

Tatsuo Shimano. 2018. 漢文訓読をあらわす三つのモデルとチョムスキー階層. In 計量国語学会第六十二回大会予稿集. 計量国語学会.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Dongbo Wang, Litao Lin, Zhixiao Zhao, Wenhao Ye, Kai Meng, Wenlong Sun, Lianzhen Zhao, Xue Zhao, Si Shen, Wei Zhang, and Bin Li. 2023a. EvaHan2023: Overview of the first international Ancient Chinese translation bakeoff. In *Proceedings of ALT2023: Ancient Language Translation Workshop*, pages 1–14, Macau SAR, China. Asia-Pacific Association for Machine Translation.

Hao Wang, Hirofumi Shimizu, and Daisuke Kawahara. 2023b. Kanbun-LM: Reading and translating classical Chinese in Japanese methods by language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8589–8601, Toronto, Canada. Association for Computational Linguistics.

Pengyu Wang and Zhichen Ren. 2022. The uncertainty-based retrieval framework for ancient chinese cws and pos. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 164–168.

Yuting Wei, Qi Meng, Yuanxing Xu, and Bin Wu. 2025. TEACH: A contrastive knowledge adaptive distillation framework for classical Chinese understanding. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3537–3550, Vienna, Austria. Association for Computational Linguistics.

Youheng W. Wong, Natalie Parde, and Erdem Koyuncu. 2024. Humanistic buddhism corpus: A challenging domain-specific dataset of English translations for classical and Modern Chinese. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8406–8417, Torino, Italia. ELRA and ICCL.

Koichi Yasuoka. 2018. 漢文の依存文法解析と返り点の関係について. In 日本漢字学会第*1*回研究大会予稿集, pages 33–48. 日本漢字学会.

Koichi Yasuoka. 2020. 漢文の依存文法解析にもとづく自動訓読システム. 日本漢字学会第*3*回研究大会予稿集, pages 60–73.

Koichi Yasuoka, Christian Wittern, Tomohiko Morioka, Takumi Ikeda, Naoki Yamazaki, Yoshihiro Nikaido, Shingo Suzuki, Shigeki Moro, and Kazunori Fujita. 2022. Designing universal dependencies for classical chinese and its application. 情報処理学会論文誌, 63:355–363.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Zhihan Zhang, Wenhao Yu, Mengxia Yu, Zhichun Guo, and Meng Jiang. 2023. A survey of multi-task learning in natural language processing: Regarding task relatedness and training methods. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 943–956, Dubrovnik, Croatia. Association for Computational Linguistics.

Bo Zhou, Qianglong Chen, Tianyu Wang, Xiaomi Zhong, and Yin Zhang. 2023. WYWEB: A NLP evaluation benchmark for classical Chinese. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3294–3319, Toronto, Canada. Association for Computational Linguistics.

A Hyperparameters

The follow table describes hyperparameters we used when finetuning pretrained language models.

Hypermeter	Value
batch size	64
optimizer	AdamW
beta1	0.9
beta2	0.999
epsilon	1e-8
weight decay	0.01
learning rate	5e-5
dropout	0.3
epoches	15
early stopping epoch	1

Table 5: Hyperparameters of finetuning pretrained language models

B LLM Prompts

We use the following prompt to make LLMs revise the POS tags generated by GiNZA:

Prompt

あなたは日本語の専門家です。次は漢文、書き下し文及び書き下し文の品詞タグ付きです。これに基づき、書き下し文の品詞タグを訂正してください。GiNZA品詞体系を使ってください。動詞と名詞の語幹及び送り仮名をに注意してください。助詞、例えば「て」、助動詞、例えば「し」に注意してください。再読文字、例えば「未だ」「将に」「須らく」「若お」「蓋ぞ」、とそれらの送り仮名に注意してください。形式名詞に対して、「名詞-普通名詞-形式名詞」のタグを使ってください。

<Classical Chinese Sentence>
<Japanese Translation>

We use the following zero-shot prompt and few-shot prompt to let LLMs generate Japanese translation and **Kaeriten** marks towards provided classical Chinese sentences:

Zero-shot Prompt

あなたは日本語と古典中国語の専門家です。次の漢文を日本語の書き下し文に翻訳してください。翻訳すると共に、一つ一つ漢字と句読点に付く返り点を書いてください。返り点を付く必要がない場合、empty string ""を使てください。また、複数の返り点が現れる場合、一つのstringに書いてください。例えば、"一"と"レ"を"一レ"に書きます。漢字の振り仮名を付けないでください。

次の漢文を読んで書き下し文と返り点を書いてください。
<classical Chinese sentence>。この漢文には<the length of classical Chinese sentence>の漢字と句読点があります。それぞれの漢字と句読点に対して返り点を提出してください。

Few-shot Prompt

あなたは日本語と古典中国語の専門家です。次の漢文を日本語の書き下し文に翻訳してください。翻訳すると共に、一つ一つ漢字と句読点に付く返り点を書いてください。返り点を付ける必要がない場合、必ずempty string ""を使てください。また、複数の返り点が現れる場合、一つのstringに書いてください。例えば、"一"と"レ"を"一レ"に書きます。漢字の振り仮名を付けないでください。

例えば、漢文「如聞泣幽咽」の書き下し文は「泣きて幽咽するを聞くが如し」で、それぞれの漢字と句読点に対する返り点は"レ", "二", "", "", "一"です。
漢文「子曰、富與貴、是人之所欲也。」の書き下し文は「子曰く、富と貴きとは、是れ人の欲する所なり。」で、それぞれの漢字と句読点に対する返り点は"", "", "", "", "", "", "", "", "", "", "レ", "", "", ""です。
漢文「罰不遷列、欲民速覩爲不善之害也。」の書き下し文は「罰、列を遷さざるは、民の速やかに不善を為すの害を覩んことを欲すればなり。」で、そ

れぞれの漢字と句読点に対する返り点は"", "レ", "レ", "", "", "二", "", "", "下", "乙", "", "甲", "", "上", "一", ""です。

次の漢文を読んで書き下し文と返り点を書いてください。
<classical Chinese sentence>。この漢文には<the length of classical Chinese sentence>の漢字と句読点があります。それぞれの漢字と句読点に対して返り点を提出してください。

C An Example of the Kaeriten Marks and the Execution of the PDA

We use the following sentence as the example:

人	君	無	以	三	寶	借	人
rén	*jūn*	*wú*	*yǐ*	*sān*	*bǎo*	*jìe*	*rén*
people	ruler	don’t	take	three	treasures	lend	other people

The **Kaeriten** marks of this sentence is: 人君無$_{二}$以$_{下}$三宝$_{上}$借$_{一レ}$人
The process of this sentence is listed below:

1. reverse the adjacent characters with レ mark: 人君無$_{二}$以$_{下}$三宝$_{上}$人借$_{一}$
2. put the character with 下 mark to the right of the character with 上 mark: 人君無$_{二}$三宝以人借$_{一}$
3. shift the character with 二 mark to the right of the character with 一 mark: 人君三宝以人借無

Finally, we get the Japanese reading order of these characters in the classical Chinese sentence. The gold standard Japanese translation of this sentence is 人君は三宝を以て人に借す無かれ. It confirms the validity of **Kaeriten** mark system.

The following table 6 demonstrates the correct execution path of the PDA. The input of this PDA is a sequence of characters with their **Kaeriten** marks. For those characters without **Kaeriten** annotated, we regard their **Kaeriten** as the special symbol E. In this case, the input sequence is ["人", E, "君", E, "無", "二", "以", "下", "三", E, "宝", "上", "借", "一", "レ", "人", E].

D The Process of *Saidoku* (Read Again) Characters

Saidoku (read again) characters are a special type of words in the classical Chinese translation. They are first read as a regular word without **Kaeriten** marks, then read as the word with **Kaeriten** annotated. For example, the following sentence annotated with **Kaeriten** marks 鼎之軽重、未$_{レ}$可$_{レ}$問也。 has Japanese translation 鼎の軽重、未だ問う可からざるなり。 In this sentence, 未だざる are **Saidoku** characters. Its Chinese character is first read as a regular character. Therefore, the characters 未だ appear following the punctuation 、. The rest part is then read as a word with **Kaeriten** mark レ. ざる shows after Chinese character 可 in the translation. The following algorithm 1 describes how we process **Saidoku** (read again) characters. In the execution of our proposed PDA, we could simply enforce the immediate output of characters when it reads **Saidoku** characters.

E Ablation Experiment Results

The following three tables 7, 8 and 9 show the experimental results of our models with 0, 1, 2, 3, 4 auxiliary tasks. We ran 16 models on the base model selected from sikubert, bert-ancient-chinese and bert-base-japanese-char.

State	Input	Stack	Transduced	Transition
q_0	["人", $E, \ldots$]	[Z_0]	""	(1)
q_0	[E, "君", $\ldots$]	["人", Z_0]	""	(4)
q_1	["君", $E, \ldots$]	[Z_0]	"人"	(8)
q_0	["君", $E, \ldots$]	[Z_0]	"人"	(1)
q_0	[E, "無", $\ldots$]	["君", Z_0]	"人"	(4)
q_1	["無", "二", $\ldots$]	[Z_0]	"人君"	(8)
q_0	["無", "二", $\ldots$]	[Z_0]	"人君"	(1)
q_0	["二", "以", $\ldots$]	["無", Z_0]	"人君"	(5)
q_0	["以", "下", $\ldots$]	["二", "無", Z_0]	"人君"	(1)
q_0	["下", "三", $\ldots$]	["以", "二", "無", Z_0]	"人君"	(5)
q_0	["三", $E, \ldots$]	["下", "以", "二", "無", Z_0]	"人君"	(1)
q_0	[E, "宝", $\ldots$]	["三", "下", "以", "二", "無", Z_0]	"人君"	(4)
q_1	["宝", "上", $\ldots$]	["下", "以", "二", "無", Z_0]	"人君三"	(10)
q_0	["宝", "上", $\ldots$]	["下", "以", "二", "無", Z_0]	"人君三"	(1)
q_0	["上", "借", $\ldots$]	["宝", "下", "以", "二", "無", Z_0]	"人君三"	(6)
q_2	["借", "一", $\ldots$]	["上", "宝", "下", "以", "二", "無", Z_0]	"人君三"	(11)
q_2	["借", "一", $\ldots$]	["下", "以", "二", "無", Z_0]	"人君三宝"	(12)
q_0	["借", "一", $\ldots$]	["二", "無", Z_0]	"人君三宝以"	(1)
q_0	["一", "レ", $\ldots$]	["借", "二", "無", Z_0]	"人君三宝以"	(6)
q_0	["レ", 人", $\ldots$]	["一", "借", "二", "無", Z_0]	"人君三宝以"	(3)
q_0	["人", E]	["レ", "一", "借", "二", "無", Z_0]	"人君三宝以"	(1)
q_0	[E]	["人", "レ", "一", "借", "二", "無", Z_0]	"人君三宝以"	(4)
q_1	[]	["レ", "一", "借", "二", "無", Z_0]	"人君三宝以人"	(9)
q_2	[]	["一", "借", "二", "無", Z_0]	"人君三宝以人"	(11)
q_2	[]	["二", "無", Z_0]	"人君三宝以人借"	(13)
q_0	[]	[Z_0]	"人君三宝以人借無"	(15)
q_4	[]	[Z_0]	"人君三宝以人借無"	-

Table 6: An example of the PDA execution.

Algorithm 1 Reading **Saidoku** Characters

```
procedure READING CHINESE CHARACTERS(Input Sequence $I$, Output Sequence $O$, PDA $A$)
    while $A$ does not terminate do
        if $A$ accepts a new input character then          ▷ steps consume a new input character
            $c \leftarrow A[0]$                                ▷ get the input character
            $A \leftarrow A[1:]$
            if $c$ is a Saidoku character then
                $O$.append($c$[regular part])
                Execute $A$ to the next step with $c$[Saidoku part] and $O$
            else
                Execute $A$ to the next step with $c$ and $O$
            end if
        else                                            ▷ steps do not consume a new input character
            Execute $A$ to the next step with $O$
        end if
    end while
    return $O$
end procedure
```

Tasks	BLEU	chrF	BERTScore	ROUGE-L	RIBES	TER	Kendall's τ	PMR	Pass Rate
Kundoku marks	62.99	57.49	94.33	84.91	56.98	21.01	96.56	93.78	93.76
+seg	64.40	58.33	94.48	85.34	57.13	20.51	96.82	94.53	92.37
+pos	62.54	58.14	94.16	84.71	56.69	20.86	96.58	93.73	93.55
+arc	65.16	60.63	**94.70**	**86.26**	**60.82**	19.38	96.76	94.22	92.58
+type	**65.68**	60.54	94.68	86.01	59.80	19.62	96.95	94.20	93.87
+seg+pos	63.59	57.77	94.27	85.07	56.49	20.90	96.42	94.20	93.01
+seg+arc	63.49	57.59	94.33	85.02	55.92	20.97	**97.01**	**94.55**	91.94
+seg+type	63.47	56.52	94.28	84.94	54.10	21.34	96.76	94.08	93.01
+pos+arc	64.33	59.67	94.49	85.57	58.98	19.98	96.37	93.84	92.90
+pos+type	65.09	**60.96**	94.65	86.11	60.52	**19.29**	96.49	94.44	91.94
+arc+type	62.57	58.23	94.29	85.09	59.46	20.97	96.34	93.52	92.47
+seg+arc+type	61.59	56.91	94.11	84.82	56.99	21.34	96.32	93.31	91.29
+seg+pos+type	63.00	57.70	94.41	85.19	56.95	20.93	96.32	93.33	**94.62**
+seg+pos+arc	62.09	57.82	94.23	85.18	58.48	20.94	96.56	94.07	90.65
+pos+arc+type	61.55	57.13	94.15	84.98	57.59	21.07	96.70	94.22	93.44
+all	61.84	57.54	94.31	85.15	58.54	20.94	96.54	94.31	91.83

Table 7: Experimental results on sikubert.

Tasks	BLEU	chrF	BERTScore	ROUGE-L	RIBES	TER	Kendall's τ	PMR	Pass Rate
Kundoku marks	64.27	57.92	94.33	84.91	55.34	20.90	96.72	94.17	**95.27**
+seg	63.76	59.30	94.35	85.51	58.95	**19.13**	96.90	94.70	90.65
+pos	65.52	61.13	**94.76**	**86.36**	60.78	19.28	**97.05**	**94.89**	92.89
+arc	63.80	58.15	94.32	85.08	55.81	20.58	96.38	93.88	93.44
+type	64.38	59.94	94.46	85.56	59.62	20.01	96.92	94.08	93.76
+seg+pos	64.46	58.21	94.36	85.46	56.26	20.57	96.78	94.79	91.07
+seg+arc	65.21	60.08	94.65	85.96	59.44	19.67	96.93	94.75	93.01
+seg+type	**65.58**	60.33	94.62	86.06	59.32	19.45	96.87	94.51	93.87
+pos+arc	64.86	59.60	94.55	85.62	58.63	20.06	96.54	94.00	93.98
+pos+type	63.45	58.43	94.35	85.22	58.79	20.67	96.62	94.19	94.09
+arc+type	65.43	**60.91**	94.68	86.19	**61.42**	**19.13**	96.90	94.71	91.29
+seg+arc+type	62.58	57.98	94.22	84.87	57.19	20.73	96.29	93.70	93.01
+seg+pos+type	64.97	60.40	94.58	86.13	59.85	19.36	96.97	94.50	92.04
+seg+pos+arc	62.61	58.73	94.32	85.17	58.80	20.49	96.56	93.95	92.37
+pos+arc+type	61.94	57.39	94.17	84.85	56.93	21.12	96.69	93.94	91.61
+all	62.46	57.16	94.14	84.99	57.21	20.91	96.88	94.65	92.04

Table 8: Experimental results on bert-ancient-chinese.

Tasks	BLEU	chrF	BERTScore	ROUGE-L	RIBES	TER	Kendall's τ	PMR	Pass Rate
Kundoku marks	52.34	47.50	92.52	79.82	46.75	27.53	93.71	88.75	87.85
+seg	53.96	50.35	92.72	80.52	50.67	26.55	93.60	88.99	91.08
+pos	54.58	49.93	92.78	80.69	49.69	26.27	93.47	88.92	**91.72**
+arc	53.48	48.77	92.66	80.36	48.46	27.06	94.15	89.86	89.57
+type	53.58	49.85	92.78	80.86	50.82	26.17	94.51	89.72	85.59
+seg+pos	54.16	49.01	92.77	80.50	47.93	27.02	93.64	88.45	91.18
+seg+arc	52.99	49.27	92.71	80.36	49.71	26.74	93.57	89.09	88.17
+seg+type	53.93	49.79	92.75	80.76	49.15	26.49	93.90	89.43	90.54
+pos+arc	53.96	50.62	92.76	80.93	**52.17**	26.20	94.51	**90.09**	89.25
+pos+type	**55.35**	**50.65**	**92.97**	**81.29**	50.58	**25.74**	**94.54**	90.04	88.92
+arc+type	53.04	48.96	92.68	80.40	48.49	26.54	94.29	89.84	90.86
+seg+arc+type	51.74	48.83	92.49	80.29	50.04	26.74	93.87	89.19	86.34
+seg+pos+type	52.33	49.05	92.68	80.60	50.01	26.72	93.48	89.28	89.46
+seg+pos+arc	53.97	49.35	92.7	80.42	48.58	26.90	93.91	89.52	89.36
+pos+arc+type	51.26	47.48	92.37	79.65	48.44	27.35	93.71	89.04	91.40
+all	52.29	47.48	92.60	79.85	46.40	27.37	93.62	89.18	90.11

Table 9: Experimental results on bert-base-japanese-char.