WakaGPT: Classical Japanese Poem Generator

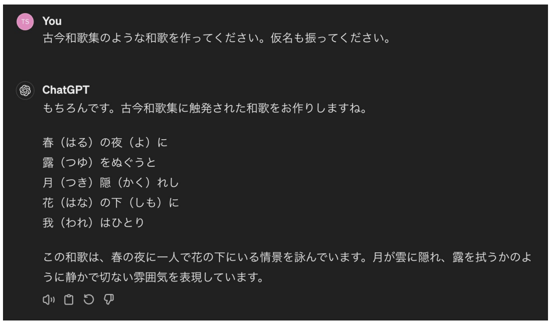
Ruixuan Tu
Department of Computer Sciences
University of Wisconsin–Madison
Madison, WI, USA
ruixuan@cs.wisc.edu

Figure 1: Example of Waka: Kokinshu 0169

あききぬと－めにはさやかに－みえ
ねとも－かせのおとにそ－おとろか
れぬる

a ki ki nu to - me ni wa sa ya ka ni - mi e
ne to mo - ka ze no o to ni zo - o to ro ka
re nu ru

Figure 2: ChatGPT4 Waka Generation Example



You
古今和歌集のような和歌を作ってください。仮名も振ってください。

ChatGPT
もちろんです。古今和歌集に触発された和歌をお作りしますね。

春（はる）の夜（よ）に
露（つゆ）をぬぐうと
月（つき）隠（かく）れし
花（はな）の下（しも）に
我（われ）はひとり

この和歌は、春の夜に一人で花の下にいる情景を詠んでいます。月が雲に隠れ、露を拭うかのように静かで切ない雰囲気を表現しています。

Abstract

Waka is a traditional Japanese poem that is usually in a certain mora sequence format. However, generating waka is challenging due to lack of data in classical Japanese and this kind of poetry, as well as the usual format restrictions. In this paper, we present WakaGPT, a waka composer based on Japanese GPT2 and the base models it is fine-tuned on. By self-supervised and semi-supervised training, we are able to generate waka poems with correct grammar and format.

1 Introduction

Waka is a traditional Japanese poem with 31 moras with 5-7-5-7-7 pattern. In linguistics, mora is a basic timing unit in the phonology of some spoken languages; it is a kana unit in Japanese. We take the poem in Figure 1 as an example:

The poetry is written in classical Japanese around 10th century, not modern Japanese. However, the existing Japanese language models (LMs); such as Kyoto University DeBERTa (Murawaki, 2023), Kyoto University GPT2 (Ueda, 2023), Waseda University RoBERTa (Kawara and Wang, 2022), and Japanese StableLM (Lee et al., 2023); are trained on modern Japanese, which makes it difficult to process classical Japanese. For example, the word conjugations, the grammar rules, and the vocabulary[1] are different. In the same way, we cannot say that there exists a good tokenizer beyond the 2-gram UniDicS (Ogiso et al., 2012), and there is no universal tokenizer that works for both modern and classical Japanese simultaneously.

We then turn to use some general-purpose LMs, including the SOTA model ChatGPT4 (OpenAI, 2024). However, we have attempted to prompt it in Japanese “Please generate a waka poetry like in Kokinshu collection, and provide the kana reading.”, but the model generates the following poem, which is in 5-7-6-6-6, far from the expected standard 5-7-5-7-7 pattern, as shown in Figure 2.

WakaVT (Takeishi et al., 2022) is the first model which uses Transformer and VAE to generate waka, using the same dataset used to fine-tune WakaGPT. The structure is carefully designed to take care of keywords for conditional generation and used additive masks to keep the pattern of the poem. In comparison, WakaGPT does not specifically design any structure, does not extracting any keyword, and neither it uses tokenization by morphological analysis tool.

There are methods to translate classical Chinese (Kanbun) to classical Japanese using encoder-based reordering and decoder-based generation (Wang et al., 2023), and to translate classical Japanese to modern Japanese (Usui and Komiya, 2023) using

[1] Some words in modern Japanese are borrowed from western languages in the future, from the aspect of the classical Japanese.

T5. To compare, our paper covers all styles of classical Japanese including those not in translation of classical Chinese and not having translation, and more foundational tasks in classical Japanese.

As the result, instead of using existing models designed for modern languages, we propose to pre-train more to transfer the knowledge of modern Japanese to classical Japanese.

In this paper, we will discuss our pre-trained language model and waka generation by using that pre-trained model. Specifically, this paper makes the following contributions to this task. Also, our source code and models are available at https://github.com/TURX/classical-lm.

1. **CHJ-GPT2/DeBERTa**: We pre-train the base causal and masked LMs for classical Japanese. These models could further be fine-tuned for various tasks in classical Japanese.
2. **WakaGPT**: We fine-tune CHJ-GPT2 on the waka dataset, and this application proves the effectiveness of pre-trained models in classical Japanese for smaller-scale tasks.

2 Language Models

2.1 Datasets

For pre-training, we use the Corpus of Historical Japanese (CHJ) (National Institute for Japanese Language and Linguistics, 2022), specifically the Nara Peiod Series (710-794) and the Heian Period Series (794-1185), reflecting the historical periods of Japan when classical Japanese was prevalent, as well as the periods of classical waka poetry collections. One line of the dataset from search queries; consisting of prior context (19 words), key word, and posterior context (19 words), as in Figure 3. We have 1453767 rows before data augmentation.

To augment the data due to the limited data from 7th century to 10th century, the earliest possible historical text records, we use different forms of writing (Chinese characters for symbol and kana characters for pronunciation) to replace the key word, forming 0-2 extra copies of each context. For the above example, we can replace the key word by its kana reading うごかし (*u go ka shi*) and ウゴカシ (*u go ka shi*).

Figure 3: Example of CHJ Dataset Row: Excerpt from The Kiritsubo Chapter, The Tale of Genji

Prior Context: 更衣 | たち | は | まして | やすから | ず |。 # 朝夕 | の | 宮仕 | に | つけ | て | も |、 | 人 | の | 心 | を | のみ |

Key Word: 動かし (*u go ka shi*)

Posterior Context: |、 | 恨み | を | 負ふ | つもり | に | や | あり | けん |、 | いと | あつしく | なり | ゆき |、 | もの | 心細げ | に | 里

Figure 4: Example of Augmented Prompt Row for WakaGPT: Kokinshu 0001

[詞書] ふるとしに春たちける日よめる\n[作者] 元方 在原元方 (169)\n[仮名] としのうちに–はるはきにけり–ひととせを–こそとやいはむ–ことしとやいはむ

For fine-tuning, we use the Waka Poetry Database (International Research Center for Japanese Studies, 2002) containing all imperial anthologies and major private collections. Associated with each poem text without line separation, there are optional metadata fields including collection name, collection date, collection place, poem preface, poem author, and poem kana reading with 5-7-5-7-7 separation. However, for the training objective, we reformulate every row with prompt tags consisting of optional Meta tags and one Poem tag within [詞書 *Meta-Preface*], [作者 *Meta-Author*], [仮名 *Poem-Kana*], [原文 *Poem-Original*], and [整形 *Poem-Aligned*].

This dataset is even more limited than the CHJ dataset by its subset nature, since waka is just one form of classical Japanese. Also due to the nature of interchangibility of Meta prompt tags, we can augment the data by different combinations and orders of prompt tags, formulating at most $(1 \cdot 0! + 2 \cdot 1! + 1 \cdot 2!) \cdot 3 = 15$ rows for every original row. We can look at an example of an augmented prompt row to understand the structure in Figure 4. We have 208972 rows before data augmentation and 722765 rows after data augmentation.

2.2 Models

Based on Kyoto University GPT2 (Ueda, 2023) and Kyoto University DeBERTa (Murawaki, 2023), we pre-train the causal LM CHJ-GPT2 and the masked LM CHJ-DeBERTa on the CHJ dataset to learn classical Japanese. We then fine-tune CHJ-

Table 1: Hyperparameters of Language Models

Model	**CHJ-GPT2**	**WakaGPT**	**CHJ-DeBERTa**
Train/Test Split	0.9/0.1	0.9/0.1	0.9/0.1
Batch Size	32	32	2048
Epochs	1	1	1
Warmup Ratio	0.1	0.1	0.1
Learning Rate	3e-4	2e-5	2e-4
LR Scheduler	Cosine	Linear	Linear
Weight Decay	0.01	0.01	0.01
Optimizer	AdamW	AdamW	AdamW
Adam β_1	0.9	0.9	0.9
Adam β_2	0.95	0.999	0.999
Adam ϵ	1e-8	1e-8	1e-8
DDP Runtime	1d 4h 2m	7h 3m	6h 3m
GPU	8x A30	2x A30	8x 2080Ti

GPT2 on the waka dataset to learn the task of waka generation as WakaGPT. The hyperparameters for the LMs are shown in Table 1. To adapt to variant GPU resources, we use Distributed Data Parallel for training on multiple GPUs, and maintain the same batch size on low memory GPUs by gradient accumulation.

2.2.1 Pre-trained Models

We recall that causal LM predicts one token based on all previous tokens but masked all future tokens, which enables text generation by greedy or beam search, as well as the self-supervisied objective of completing the masked tokens from the unlabeled contexts and calculating the resulted cross-entropy loss. On the other hand, a masked LM predicts randomly masked tokens and generates bidirectional self attention, which enables classification tasks based on tokens, without changing the description of the self-supervised objective.

Kyoto University GPT2 is a causal LM on modern Japanese, corresponds to medium size (335M) of parameters, and is at character level. Since the tokenization rule changes a lot between modern and classical Japanese, it is difficult to freeze a tokenizer for training. From the character level tokenization, we no longer need any tokenization rule (like morphological analysis tool), where we leverage morpheme understanding burden on the model itself. In this way, it is easier to transfer the conditional distributions from modern Japanese to classical Japanese, since most grammar shifts happen at the character level, without requiring changing every related subword associated with specific grammars.

On the other hand, Kyoto University DeBERTa is a masked LM on modern Japanese, corresponds to base size (137M) of parameters, is at character level, and applies whole word masking. We use the same character level tokenization for the same reason as GPT2. However, specifically for bidirectional attention, we need to mask a portion of the text, while in this setting, we could mask the whole word. In English BERT, it makes the mask prediction task more challenging, as originally the model can see a part of subwords since the whole word might not be completely masked. In Chinese-Japanese-Korean languages, it significantly help the model to learn the word boundaries in character level, since there is no space between words in these languages, as shown by Cui et al. (2021) for Chinese case. We believe this setting could also apply on classical Japanese that also requires knowledge transfer of word boundaries.

For our pre-trained models, we use the same self-supervised objectives as we picked on the base models. We predict the masked tokens and calculate loss compared to the original tokens under the mask. By our data augmentation method in the CHJ dataset, the model could learn the correspondence between different forms of writing, namely the pronunciation/reading of words not written entirely in kana, which is also useful in generating waka poems entirely in kana.

2.2.2 WakaGPT: Waka Poem Composer

As mentioned in Section 2.1, we have created prompt labels for fine-tuning CHJ-GPT2 as WakaGPT. The objective of this task is that the model should generate or complete the whole poem, given optionally preface, author, and/or leading lines. This formation allows direct application of the generative objective of causal LM, so we are able to change only the dataset for continuation of the training.

As for the prompt design, it worth highlighting that we leave the poem tag at the end of the prompt, so it matches the completion given by all prior conditions, and the model would learn to end the poem after generating for the last tag. Moreover, we still stick with fine-tuning, since it is difficult to learn formation of the poem structure from in-context learning (Dong et al., 2023).

A demo of WakaGPT is available at HuggingFace Space.

We then comment on a generated waka by WakaGPT as in Figure 5. This poem is grammarly correct, uses the correct vocabulary in the period, and follows the 5-7-5-7-7 pattern. However, it connects the two seasons (autumn and spring with cherry blossoms), which is unusual in classical waka, and

Figure 5: Example of WakaGPT Generation

Condition Given: First Line あききぬと–

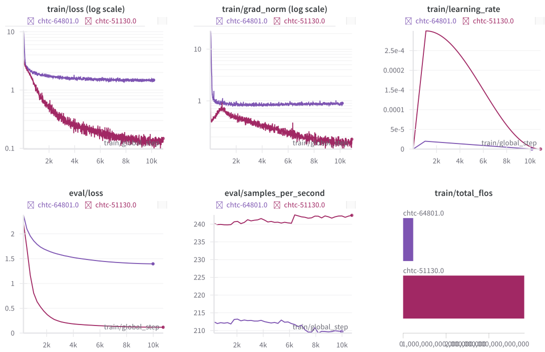
Generated: あききぬと–おもひける かな–やまさくら–さきてちりぬる –はなのしたかせ

Pronunciation: *a ki ki nu to / o mo i ke ru ka na / ya ma za ku ra / sa ki te chi ri nu ru / ha na no shi ta ka ze*

Human Assigned Original Form: 秋きぬと思ひけるかな山桜咲きて散りぬる花の下風

Human Translation: When autumn came / I recalled / the mountain cherry blossoms / bloomed and then scattered / flowers and the breeze below.

Figure 6: Training Logs of GPT2. Run IDs 51130: CHJ-GPT2, 64801: WakaGPT.



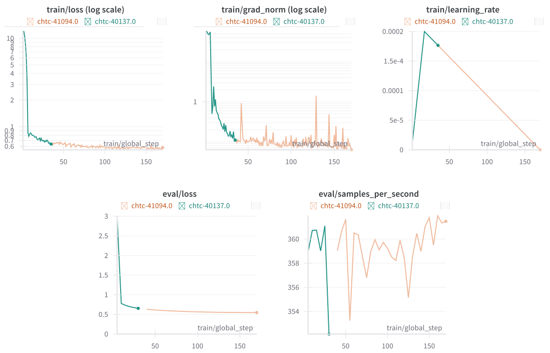
it is hard to evaluate the conceit of the breeze in the last line.

2.2.3 Training Details

As in Figure 6, we train the GPT models (CHJ-GPT2 and WakaGPT) on their respective datasets. We can see the training losses of both models converge under regularization. There is no significant overfitting: values of train loss and evaluation loss are very close. CHJ dataset costs slightly less time to forward than Waka dataset. However, CHJ dataset is much larger, taken 10 times of FLOS to train this one epoch.

As in Figure 7, we train the DeBERTa model (CHJ-DeBERTa) on the CHJ dataset. The loss and gradient norm still converges to lower values without sign of overfitting, and we observe much faster forward speed which is from the smaller number of parameters, which also results in faster training time.

Figure 7: Training Logs of DeBERTa. Keep training the same model.



3 Discussion

We have presented our language models CHJ-GPT2 and WakaGPT for classical Japanese. We successfully prove the effectiveness of pre-trained models on pure text with limited labeled data by the demos we provide. In this way, we provide a better yet near-domain data demanding method than data augmentation to improve the performance of models on low-resource languages, which we applies on classical Japanese. In terms of real-world applications, WakaGPT could be used to generate waka poems for educational purposes in K-12 and college curriculum in Japan, and in libraries and museums.

Take classical Japanese as an example in this paper, our method should also applies to other low-resource languages, especially for low resource situation in limited labeled dataset but larger unlabeled text and image records, and/or when we can apply a transition on a larger family of languages; for example, dialectal Arabic and classical Chinese. Our method could also expedite the creation of better and larger language resources with less labor by extending existing datasets.

4 Limitations

The CHJ dataset contains a continuum of data from Nara Peiod (710-794) to Taisho Period (1912-1926), so it may help the model to learn the retrogression from modern Japanese to classical Japanese in better granularity, and this could bring much more data involved in the training. However, we only use the Nara and Heian Periods in this work, so it is more abrupt for the model to learn the transition, but this solution fits our computing resources and also proves our pre-training method. Still, we have not yet tested another pre-trained model of DeBERTa on applications.

From Figure 6, the learning rate of WakaGPT seems not be sufficient, resulting in larger grad norm and loss than CHJ-GPT2, indicating a potential underfitting. Underfitting might also exist in other models since we train only one epoch due to limited computing resources. Therefore, our work here is more of a proof of concept, instead of unleashing the full potential of the models with multiple train epochs. For prompt design of WakaGPT, we could also consider styles of tags other than this one-bracket style, such as XML/HTML style with a close tag by slash. However, we are not experimenting this for potentially better performance as introduced by Aghajanyan et al. (2021), since the current style is already working, and our data for CHJ-GPT2 pre-training does not involve XML/HTML tag.

Acknowledgements

This work was done using computing resources provided by University of Wisconsin–Madison Center for High Throughput Computing (2006).

References

Armen Aghajanyan, Dmytro Okhonko, Mike Lewis, Mandar Joshi, Hu Xu, Gargi Ghosh, and Luke Zettlemoyer. 2021. HTLM: Hyper-Text Pre-Training and Prompting of Language Models. ArXiv:2107.06955 [cs].

Center for High Throughput Computing. 2006. Center for High Throughput Computing.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-Training with Whole Word Masking for Chinese BERT. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3504–3514. ArXiv:1906.08101 [cs].

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A Survey on In-context Learning. ArXiv:2301.00234 [cs].

International Research Center for Japanese Studies. 2002. Waka Poetry Database. Version 2022.3, Chunagon Version 2.5.2.

Daisuke Kawara and Hao Wang. 2022. nlp-waseda/roberta-base-japanese. HuggingFace Model Hub commit 49ce73e.

Meng Lee, Fujiki Nakamura, Makoto Shing, Paul McCann, Takuya Akiba, and Naoki Orii. 2023. Japanese StableLM Base Alpha 7B. HuggingFace Model Hub commit e6cc3ee.

Yugo Murawaki. 2023. ku-nlp/deberta-v2-base-japanese-char-wwm. HuggingFace Model Hub commit 29498bb.

National Institute for Japanese Language and Linguistics. 2022. Corpus of Historical Japanese. Version 2022.3, Chunagon Version 2.5.2.

Toshinobu Ogiso, Mamoru Komachi, Yasuharu Den, and Yuji Matsumoto. 2012. UniDic for Early Middle Japanese: a Dictionary for Morphological Analysis of Classical Japanese. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 911–915, Istanbul, Turkey. European Language Resources Association (ELRA).

OpenAI. 2024. GPT-4 Technical Report. Issue: arXiv:2303.08774 arXiv: 2303.08774 [cs].

Yuka Takeishi, Mingxuan Niu, Jing Luo, Zhong Jin, and Xinyu Yang. 2022. WakaVT: A Sequential Variational Transformer for Waka Generation. *Neural Processing Letters*, 54(2):731–750.

Nobuhiro Ueda. 2023. ku-nlp/gpt2-medium-japanese-char. HuggingFace Model Hub commit 8a49a32.

Hisao Usui and Kanako Komiya. 2023. Translation from Historical to Contemporary Japanese Using Japanese T5. In *Proceedings of the Joint 3rd International Conference on Natural Language Processing for Digital Humanities and 8th International Workshop on Computational Linguistics for Uralic Languages*, pages 27–35, Tokyo, Japan. Association for Computational Linguistics.

Hao Wang, Hirofumi Shimizu, and Daisuke Kawahara. 2023. 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.