

WakaGPT: Japanese Waka Poem Composer

Ruixuan Tu (ruixuan@cs.wisc.edu)

In-class Presentation, STAT 453 Spring 2024, University of Wisconsin–Madison

2 May 2024

What is Waka?

Waka is a traditional Japanese poem with 31 moras in 5-7-5-7-7 pattern.

Mora: In linguistics, mora is a basic timing unit in the phonology of some spoken languages; a kana unit in Japanese.

Collection of Hiragana: あいうえおかきく
けこさしすせそたちつてとなにぬねのはひ
ふへほまみむめもやゆよらりるれろわをん
(one form of kana, cursive)

Kanji: Chinese characters used in Japanese writing

Example of Kanji with Hiragana reading:
日本 (にほん, *ni ho n*)

あききぬと/めにはさやかに/見 (み) えね
とも/風 (かぜ) のおとにそ/おとろか
れぬる

a ki ki nu to (5)

me ni Fa sa ya ka ni (7)

mi e ne do mo (5)

ka ze no wo to ni zo (7)

o do ro ka re nu ru (7)

When autumn came

My eyes clearly

Could not see it, yet

In the sound of the wind

I felt it. [Source of Translation](#)

Objective

Given optionally preface sentence, author, and/or leading lines: generate/complete the whole poem.

For example, the text in gray could be generated:

Preface: 秋立つ日よめる
Composed on the first day of autumn.

Author: 敏行 藤原敏行朝臣 (018)
Fujiwara no Toshiyuki (018)

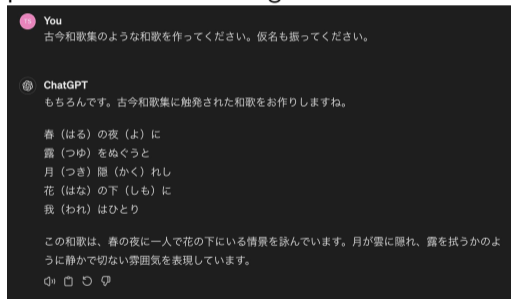
Kana: あききぬと/めにはさやかに/みえね
とも/かぜのおとにそ/おとろかれぬる
*a ki ki nu to/me ni Fa sa ya ka ni/mi e ne do
mo/ka ze no wo to ni zo/o do ro ka re nu ru*

Challenges of Waka Generation

Challenges:

- The poetry is written in classical Japanese, not modern Japanese
- It also requires to generate in the specific format
- No good tokenizer (beyond closed-source 2-gram [UniDicS](#)) available
- ChatGPT performs poorly in Waka generation

Ask ChatGPT 4 “Please generate a waka poetry like in Kokinshu collection, and provide the kana reading.”



This generated poem is in 5-7-6-6-6, far from the standard 5-7-5-7-7 pattern.

Previous Work

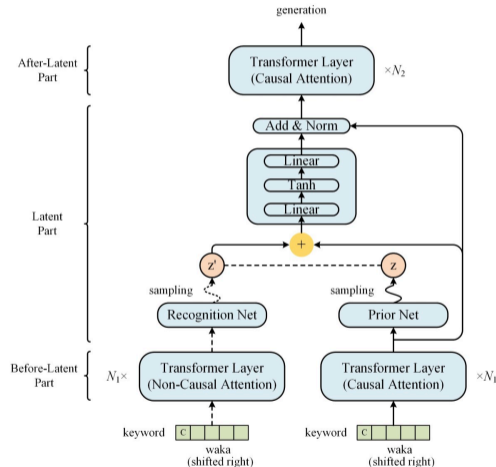
WakaVT:

- First waka composer model that use Transformer and VAE
- Trained on same waka dataset

Comparison:

- Its task is slightly different from ours: given one keyword, and generate the whole poem
- Our method is simpler without non-causal attention and MLP layers (recognition net and prior net), just the decoder (GPT)
- Our method is based on pre-training on a larger dataset

WakaVT Structure:



Novelty of Our Method: Pre-trained Model

CHJ Dataset

- **日本語歴史コーパス CHJ** (Classical Japanese Historical Corpus): Nara and Heian Periods (710-1185)
- **Row:** prior context (19 words), key word, and posterior context (19 words)
- **Size:** 1453767 rows = 3GB CSV files (before tokenization)
- **Data Augmentation:** different forms of writing (kanji for symbol and kana for pronunciation) to replace the key word
 - Form 0-2 extra copies of each context
 - Learn the reading of non-kana words

Row Example: 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*)

Augmentation Candidates: うごかし (*u go ka shi*), ウゴカシ (*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

Waka Dataset

- 和歌データベース (Waka Poetry Database)
- Row = Prompt Tags (optional Meta tags + 1 Poem tag): [詞書 Meta-Preface] [作者 Meta-Author] [仮名 Poem-Kana] [原文 Poem-Original] [整形 Poem-Aligned]
- **Data Augmentation:** different combinations and orders of prompt tags, at most $(1 \cdot 0! + 2 \cdot 1! + 1 \cdot 2!) \cdot 3 = 15$ lines
- **Size:** 208972 rows = 48.1MB Python Pickle in Pandas before augmentation; 722765 (3.46x) rows after augmentation

Preview of df[0]:

```
>>> d[0]
{'collection_date': '(延喜五年四月十八日) (905年5月24日)', 'collection_name': '古今集', 'collection_name_yomi': 'こきんしゅう', 'collection_note': '現存伝本には延喜七年や延喜十三年三月十三日の歌も収められている。官職表記は延喜十三年四月或いは延喜十年から十七年までわたるとされたことと延喜五年に成立したとしてもその後しばらくは加除訂正が行われたとみられる。', 'collection_place': '', 'poem_author': '元方 在原元方 (169)', 'poem_comment': '異同資料句番号: 00001', 'poem_id': '00001', 'poem_preface': '[詞書] ふるとしに春たちける日よめる', 'subcollection_id': 'i001-001', 'subcollection_name': '古今集 巻一: 春上', 'poem_text_type': 'kan', 'poem_text': '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', 'prompt': '[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', '_index_level_0_': 0}
```

Augmented Prompt:

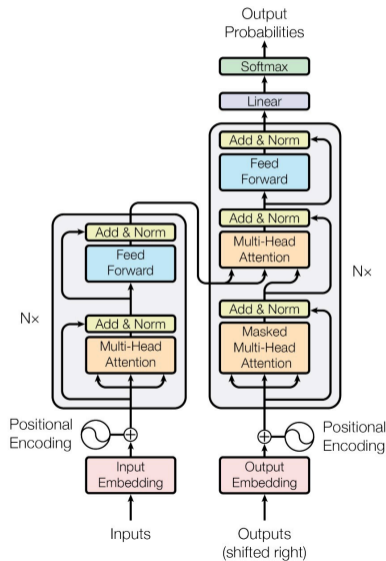
```
[詞書] ふるとしに春たちける日よめる |n
[作者] 元方 在原元方 (169) |n
[仮名] としのうちに - はるはきにけり -
ひととせを - こそとやいはむ - ことしと
やいはむ
```

Order of Training

1. **Kyoto University GPT2**: Medium size, character-level
2. CHJ-GPT2: one epoch on CHJ Dataset, based on Kyoto University GPT2
3. WakaGPT: one epoch on Waka Dataset, based on CHJ-GPT2

Character-level model: no need for tokenization; leverage morpheme understanding on model

Illustration of Transformer Structure:
Encoder at left, Decoder at right
GPT: Decoder-only Transformer
Source: [Attention Is All You Need](#)



Causal LM Training Paradigm

Pre-training: Self-supervision Objective

Causal LM: predict one token, given all previous tokens but none of the future tokens

Self-supervision: treat the input text itself as label, for unlabeled data

Generation/Completion: greedy/beam search maximal likelihood tokens one by one
Therefore, we predict the token within the input text, and mask the following tokens

Equivalent batch size is achieved by dynamic gradient accumulation and gradient checkpointing on GPUs with lower memory

Fine-tuning: Prompt Based

Prompt design: leave the poem tag at the end of the prompt, so it would be completed based on prior conditions

<i>kono</i>	<i>eiga</i>	<i>ga</i>	<i>kirai</i>	[sos]	I	hate	this	movie	[eos]
■	■	■	■	□	□	□	□	□	□
■	■	■	■	■	□	□	□	□	□
■	■	■	■	■	■	□	□	□	□
■	■	■	■	■	■	■	□	□	□
■	■	■	■	■	■	■	■	□	□
■	■	■	■	■	■	■	■	■	□

Source: [Junjie Hu's CS769](#)

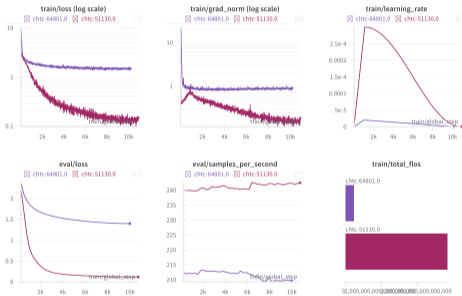
Hyperparameters

	CHJ-GPT2	WakaGPT
Train/Test Split	0.9/0.1	0.9/0.1
Batch Size	32	32
Number of Train Epochs	1	1
Warmup Ratio	0.1	0.1
Learning Rate	3e-4	2e-5
LR Scheduler	Cosine	Linear
Weight Decay	0.01	0.01
Optimizer	AdamW	AdamW
Adam β_1	0.9	0.9
Adam β_2	0.95	0.999
Adam ϵ	1e-8	1e-8
DDP Runtime	1d 4h 2m	7h 3m
GPU	8x A30 24GB (CHTC)	2x A30 24GB (CHTC)

Log Highlights

Run 51130.0: CHJ-GPT2

Run 64801.0: WakaGPT



- Not overfitting: train loss and eval loss are very close
- CHJ dataset costs slightly less time to forward than Waka dataset
- However, CHJ dataset is much larger, taken 10x FLOS to train one epoch

Limitations

- Underfitting: WakaGPT has non-sufficient (low) learning rate, resulting in larger grad norm and loss than CHJ-GPT2
- Not filtering out augmented pairs for train/eval split

Conclusions

- Training losses of both models converge under regularization

Demonstration

The model is hosted as inference endpoints on [HuggingFace Spaces](#) and [Funix Cloud](#). You can try it out at either of the following links:

- <https://huggingface.co/spaces/TURX/japanese-lm>
- <https://gpt.turx.tokyo>

The screenshot shows a web browser window with the URL gpt.turx.tokyo. The page title is "WakaGPT Poem Composer". On the left, there is a sidebar with navigation options: "Functions / Pages (3)", "Home", "Custom Prompt Japanese GPT-2", and "WakaGPT Poem Composer" (which is selected). The main content area is titled "WakaGPT Poem Composer" and contains the following text:

Generate a Japanese waka poem in 5-7-5-7-7 form using WakaGPT. A sample poem (Kokinshu 169) is provided below:

Preface: 秋立つ日よめる
Author: 敏行 藤原敏行朝臣 (018)

Kana (Kana only with Separator): あききぬと-めにはさやかに-みえぬとも-
さやかに-みえぬとも-かぜのおとにそ-おとるかれぬ

Original (Kana + Kanji without Separator): あききぬと
めにはさやかに見えぬとも風のおとにそおとるかれぬ

Aligned (Kana + Kanji with Separator): あききぬと-め
にはさやかに-見えぬとも-風のおとにそ-おとるかれぬ

Below the text are three input fields:

- Preface (Kotobagaki) in Japanese (optional)
- Author Name in Japanese (optional)
- First Line of Poem in Japanese (optional)

The first line of the poem is pre-filled: あききぬと-めにはさやかに-みえぬとも

At the bottom, there are radio buttons for "Waka Type": Kana, Original, Aligned.

At the very bottom, it says "Powered by [Funix.io](#), minimally building apps in Python".

On the right side of the interface, there is a "Prompt" field containing: [匿名] あききぬと-めにはさやかに-みえぬとも-
Mas New Tokens: 15
Removed Malformed: 0
Results: あききぬと-めにはさやかに-みえぬとも-しもにふくらぬ-た家のをやなき

Sample Generation

Prompt: first line あききぬと —

Kana form (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*

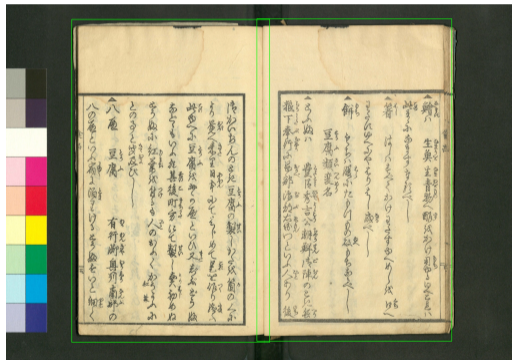
Human Assigned Original Form: 秋きぬと思ひけるかな山桜咲きて散りぬる花の下風

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

Concurrent Work

- CHJ-DeBERTa: Pre-trained masked LM (whole word masking, character-level, and bidirectional attention) on CHJ dataset for non-causal attention/classification tasks
- PageRCNN: object detection for **page boundaries** from booklet scans based on Faster R-CNN

Demo will be published along with application of these models in this summer.



Future Work

- Fix the limitations mentioned before
- Train on continuum of datasets to adapt variation: from modern, to pre-modern, and finally to classical Japanese
- Seq2Seq applications: recognition, translation, and inference

Recording

Thanks. Please check for [Zoom recording](#).