

---

# WAKAVT: A SEQUENTIAL VARIATIONAL TRANSFORMER FOR WAKA GENERATION

---

A PREPRINT

**Yuka Takeishi \***

School of Foreign Studies  
Xi'an Jiaotong University  
Xi'an, Shaanxi, P.R. China  
yuka30@stu.xjtu.edu.cn

**Mingxuan Niu \***

School of Computer Science and Technology  
Xi'an Jiaotong University  
Xi'an, Shaanxi, P.R. China  
nmx2016@stu.xjtu.edu.cn

**Jing Luo**

School of Computer Science and Technology  
Xi'an Jiaotong University  
Xi'an, Shaanxi, P.R. China  
luojing1@stu.xjtu.edu.cn

**Zhong Jin**

School of Foreign Studies  
Xi'an Jiaotong University  
Xi'an, Shaanxi, P.R. China  
jinzhongshici@aliyun.com

**Xinyu Yang**

School of Computer Science and Technology  
Xi'an Jiaotong University  
Xi'an, Shaanxi, P.R. China  
yxyphd@mail.xjtu.edu.cn

## ABSTRACT

Poetry generation has long been a challenge for artificial intelligence. In the scope of Japanese poetry generation, many researchers have paid attention to Haiku generation, but few have focused on Waka generation. To further explore the creative potential of natural language generation systems in Japanese poetry creation, we propose a novel Waka generation model, WakaVT, which automatically produces Waka poems given user-specified keywords. Firstly, an additive mask-based approach is presented to satisfy the form constraint. Secondly, the structures of Transformer and variational autoencoder are integrated to enhance the quality of generated content. Specifically, to obtain novelty and diversity, WakaVT employs a sequence of latent variables, which effectively captures word-level variability in Waka data. To improve linguistic quality in terms of fluency, coherence, and meaningfulness, we further propose the fused multilevel self-attention mechanism, which properly models the hierarchical linguistic structure of Waka. To the best of our knowledge, we are the first to investigate Waka generation with models based on Transformer and/or variational autoencoder. Both objective and subjective evaluation results demonstrate that our model outperforms baselines significantly.

**Keywords** Waka generation · self-attention mechanism · variational autoencoder · linguistic quality · novelty · diversity

---

\*Yuka Takeishi and Mingxuan Niu contribute equally to this work.

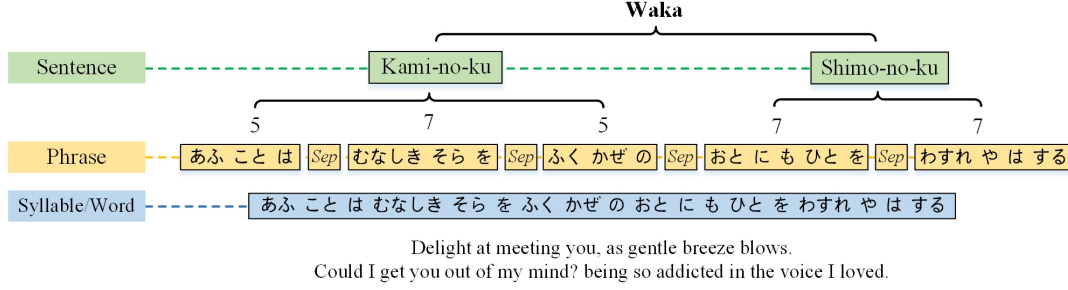


Figure 1: Hierarchical linguistic structure of Waka

## 1 Introduction

Waka<sup>2</sup> is a type of fixed verse with a long history in Japan. Rich in rhythms and lyricism, Waka can deeply express people’s thoughts and feelings. As one of the most valuable literary genres of classical Japanese literature, Waka is worthy of inheritance and development, which we believe can be further promoted by investigating Waka creation with artificial intelligence (AI). AI has demonstrated its great potential in the area of art creation, such as poetry generation [Oliveira, 2017] and music composition [Ji et al., 2020]. However, it still remains a controversial issue whether AI is able to create art like human beings, and presently, poems created by machines cannot really stand comparison with those of human poets. Poem creation is a long-term challenge for AI, while our study on Waka generation can provide reference and enlightenment for the development of this field. In addition, through building intelligent Waka generation systems, we can promote potential applications of humanizing AI in various areas such as electronic entertainment and cultural education.

Due to poetry’s attractive aesthetic value, the study of automatic poetry generation has been popular for many years. Among traditional methods, the representative ones are approaches using templates and/or rules [Colton et al., 2012, Oliveira, 2012], genetic algorithms [Manurung et al., 2012], statistical machine translation methods [He et al., 2012], and text summarization methods [Yan et al., 2013]. Traditional methods typically require experts to develop well-designed rules to remedy the defect that the models have no deep understanding of the semantic meaning. With the fast development of deep learning, neural network-based poetry generation systems have been proposed [Van de Cruys, 2020, Guo et al., 2019]. These systems deal with a large amount of linguistic features and semantic relations in poetry more automatically through well-designed network structures and sufficient training data. In the scope of Japanese poetry generation, various Haiku generation systems have been established based on traditional methods [Rzepka and Araki, 2015, Tosa et al., 2008]. In recent years, RNN language models and Generative Adversarial Networks (GAN) have been introduced into the study of Haiku generation [Hirota et al., 2018, Wu et al., 2017].

However, Waka generation is more challenging and less studied. Based on the interactive genetic algorithm, a Waka generation system which is able to process Kansei information given texts from the user has been established [Yang and Hagiwara, 2016]. Masada et al. [Masada and Takasu, 2018] has proposed a scoring method based on the latent Dirichlet allocation (LDA) to select diverse Waka poems generated by the RNN model. Nevertheless, there is still no study that specifically proposes a deep generative model to generate Waka poems with high content quality, as far as we know. The reasons for this include: (1) Compared with Haiku, Waka is written in ancient Japanese and more difficult to comprehend. (2) Waka is short in length but rich in content, and can be used to express various themes (seasons, love affair, life philosophy, etc.), which raises challenges for data modeling. (3) Voiced sounds<sup>3</sup> are hard to recognize in the writing system of Waka, making it difficult to build a large and available Waka dataset of segmented words.

As is done in general poetry generation tasks, both form and content features need to be considered. In terms of form, Waka follows strict morae constraint, with a short form but a hierarchical structure (Fig. 1). Waka can be divided into five phrases following the morae pattern of 5, 7, 5, 7, 7. Generally, the first three phrases and the last two phrases are referred to as Kami-no-ku (or “upper verse”) and Shimo-no-ku (or “lower verse”), respectively. Morae constraint and hierarchical structure bring formal beauty to Waka, but also present challenges in Waka generation. In terms of

<sup>2</sup>To avoid ambiguity, our study of Waka is limited to Tanka. For an introduction to various types of Waka, see [https://en.wikipedia.org/wiki/Waka\\_\(poetry\)](https://en.wikipedia.org/wiki/Waka_(poetry)).

<sup>3</sup>In the Japanese writing system, the *dakuten* (voicing mark) is used to indicate kana characters supposed to be pronounced voiced. However, it’s not used by ancient poets in the writing of Waka, making it difficult to recognize the pronunciations. For more information, see [https://en.m.wikipedia.org/wiki/Dakuten\\_and\\_handakuten](https://en.m.wikipedia.org/wiki/Dakuten_and_handakuten).

content, the linguistic quality, novelty, and diversity of the generated poems are important factors in evaluating the capability of a poetry generation model. For linguistic quality, a generated poem is expected to follow grammatical rules, achieve semantic coherence and convey a meaningful message. As for novelty, the model should create poems itself, rather than copying or recombining pieces from human-created poems. To reflect diversity, poems generated with different inputs should be distinguishable from each other, and repetitive, generic results should be avoided. It is noteworthy that the relationship among the above factors is not simple. Novelty and diversity tend to be positively correlated since both of them reflect the imagination and inventiveness in the selection of content and the use of language. However, they are usually incompatible with linguistic quality as intriguing wording and phrasing may cause problems of linguistics. Consequently, these factors actually need to be balanced in order to get convincing results.

Waka, short in form, requires a highly condensed language to express rich content, thus novelty and diversity become prominent challenging factors. For the generation of other types of poetry, several approaches have been put forward to improve novelty or diversity. Zhang et al. [Zhang et al., 2017] combined RNN and memory networks to generate innovative Chinese poems. Li et al. [Li et al., 2018] applied CVAE to Tang poetry and Song lyrics generation to improve term novelty. Hirota et al. [Hirota et al., 2018] proposed using different datasets to pre-train the generator and the discriminator of SeqGAN to avoid plagiarism in the generation of Haiku. Shen et al. [Shen et al., 2020] proposed a novel method of polishing drafts, which improves the novelty and diversity of the language usage in modern Chinese poetry through impressive word detection mechanisms. In the above approaches, CVAE models the text by introducing a single latent variable, which is, however, insufficient to capture the high variability of texts [Du et al., 2018]. In the scope of dialogue generation and machine translation, models based on a sequence of latent variables have been proposed and proven to generate more diverse, informative texts [Du et al., 2018, Lin et al., 2020, Schulz et al., 2018]. These models introduce a latent variable for each token of the input sequence, thus capable of modeling the variability at the word level. We will show that it’s feasible to incorporate a sequence of latent variables into Waka generation models to promote their creativity.

In this paper, we propose a Waka generation model, WakaVT, which handles both the form constraint and the content quality. WakaVT is developed by incorporating a sequence of latent variables sequentially into a conditional Transformer language model [Keskar et al., 2019, Vaswani et al., 2017]. It takes a user-specified keyword as input through control codes [Keskar et al., 2019]. As for form, we propose an additive mask-based method to satisfy the morae constraint of Waka. In terms of content, WakaVT employs a sequence of latent variables to capture the high variability of Waka data to enhance the novelty and diversity of generated content. Besides, we propose the Fused Multilevel Self Attention Mechanism (FMSA) to properly model the hierarchical structure of Waka, enabling a better understanding of the linguistic features, and as a result, improving the linguistic quality of the generated poems. With no prior study for comparison, we build three baselines based on the RNN or the Transformer, considering both language model and variational autoencoder (VAE) architectures. To evaluate the generated Waka in a more comprehensive way, we put forward word-based and 5/7-morae phrase-based automated metrics for novelty and diversity. Both objective and subjective evaluation results demonstrate that WakaVT is able to create Waka poems with strong novelty and diversity, and significantly boosts the linguistic quality in terms of fluency, coherence, and meaningfulness, compared to the baselines.

In summary, our contributions are as follows:

- (1) To the best of our knowledge, this is the first report applying models based on Transformer and/or VAE to the study of Waka generation. The models we built can automatically generate Waka based on user-specified keywords.
- (2) We propose WakaVT, a novel model for Waka generation, which improves novelty and diversity of the generated results through a sequence of latent variables. Moreover, we propose the Fused Multilevel Self Attention Mechanism (FMSA) to boost the linguistic quality by making the model aware of the hierarchical structure of Waka.
- (3) We design several baselines for generating Waka with given keywords. The objective and subjective evaluations indicate that our model outperforms baseline models significantly.

The rest of this paper is structured as follows: Sect. 2 presents previous study related to the work in this paper. Sect. 3 elaborates on the WakaVT model. Sect. 4 illustrates the experimental results and analysis. Sect. 5 draws conclusions.

---

## 2 Related work

From the perspective of task, our study focuses on the controllable poetry generation given keywords. This section first introduces related studies on controllable poetry generation based on deep learning (Sect. 2.1), and then discusses research progress in Japanese poetry generation (Sect. 2.2).

### 2.1 Controllable poetry generation

Just as a human being is motivated by a certain stimulus to create poetry, a poetry generation system also needs specific prompts to purposefully generate poems. Poetry generation systems are generally conditioned on a user-provided query, so that the generation reflect, to some extent, the user’s writing intent. Different types of queries can control different aspects of poetry.

The most common query is to input words or texts to control the poetry’s content features. The text provided by the user is often used to guide the system to generate a poem of specific topics, scenarios and concepts. According to the granularity of the input control over the generation, such studies can be divided into three categories. Firstly, some researchers used the keywords to explicitly control the content of the first line of a poem [Deng et al., 2020, Zhang and Lapata, 2014]. Zhang et al. [Zhang and Lapata, 2014] expanded, combined, and selected several keywords entered by the user to obtain the first line of a Chinese poem, and then used RNN to complete it. Deng et al. [Deng et al., 2020] used the BERT-based Seq2Seq model to convert the given keywords into the first line of a Chinese poem and then completed and polished the draft. Secondly, some researchers used keywords as global information to ensure that the poetry meets the theme expected by the user [Ghazvininejad et al., 2016, Li et al., 2018]. The Hafez system [Ghazvininejad et al., 2016] adopted word2vec to calculate topically related words and phrases, and combined the Finite-state Acceptor and RNN to generate English poetry. Li et al. [Li et al., 2018] took the user-specified title as the topic, and applied a CNN-based discriminator to learn the thematic consistency between the title and each line of the poetry with adversarial training. Thirdly, to have more fine-grained control over the content, researchers used the TextRank algorithm to extract a keyword for each line of the poem from the words, sentences, or documents provided by the user [Wang et al., 2016, Yang et al., 2018a]. Wang et al. [Wang et al., 2016] proposed a planning-based method inspired by human creation outlines, in which keywords are extracted from the input text, and a sub-topic is assigned to each line of the poem in the planning phase. Yang et al. [Yang et al., 2018a] proposed a CVAE model based on a hybrid decoder that effectively enhances the sub-topic information in latent variables utilizing a deconvolutional neural network.

In addition to the above studies, researchers have also explored the possibilities of taking styles, emotions, rhetoric methods, formats, or images as queries for poetry generation. In terms of style, some poetry generation systems could generate poems of specific styles represented by labels [Yang et al., 2018b, Yi et al., 2020]. In terms of emotion, users can control the emotions expressed in poetry in different ways. For example, users write a blog containing emotional content, which is then machine-converted into English poetry to express their emotions [Misztal and Indurkha, 2014]. Users can also directly input sentiment labels into the model to control the polarity and intensity of sentiment when generating Chinese poetry [Chen et al., 2019]. In terms of rhetoric, Liu et al. [Liu et al., 2019] proposed an encoder-decoder framework which is able to control the use of metaphor and personification in modern Chinese poetry. In terms of format, several unified generation frameworks [Guo et al., 2019, Hu and Sun, 2020] support the generation of poetry of multiple genres and formats. These systems either provide users with multiple options, or define a unified input format to control different genres and formats. The SongNet model [Li et al., 2020] provides a method for fine-grained control of text formatting, where the user inputs a sequence of placeholders to specify the length, syntactical structure, rhyming scheme, or partial predefined content of the Song Lyrics or Sonnet. In addition, there are also studies on poetry generation that take image data as multi-modal inputs [Liu et al., 2018, Xu et al., 2018], attempting to create poetry related to the object, emotion, scene, or topic of an image.

Our task is to take a keyword provided by the user as a query to make the model generate a Waka poem containing this keyword. Due to the short length of Waka, users are allowed to input only one keyword, whose position in the generated poem is not limited, thus encouraging the model to create concise and novel content with the keyword as the core.

### 2.2 Japanese poetry generation

The study of Japanese poetry generation mainly focuses on Haiku, a fixed verse developed from Waka. Haiku is shorter than Waka and follows the morae pattern of 5, 7, 5. The research methods for Haiku generation are generally in three categories: approaches using templates and/or rules, genetic algorithms, and approaches using neural networks. The study on Waka generation is also discussed according to these categories.

The first approach ensure the generated Haiku is formally and grammatically correct through templates or rules, and fills the Haiku with words or phrases extracted from rich lexical resources. The template used can be either an artificially prescribed Haiku format [Tosa et al., 2008] or a syntactic pattern drawn from the Haiku corpus [Netzer et al., 2009, Rzepka and Araki, 2015]. The repository of lexical resources is built using thesauri [Tosa et al., 2008] or web blogs related to Haiku [Rzepka and Araki, 2015, Wong et al., 2008]. In addition, researchers considered various methods to ensure the generated Haiku made sense. Wong et al. [Wong et al., 2008] calculated the semantic similarity between phrases based on the vector space model (VSM) to match the three lines of a Haiku poem with each other. Netzer et al. [Netzer et al., 2009] used the Word Association Norm Network to ensure correlation between the theme words contained in the candidate lines of Haiku. Ito et al. [Ito et al., 2018] selected and arranged elements from the narrative to generate Haiku, which is naturally associated with the background story.

The genetic algorithms are inspired by the process of creating poems by human beings. Generally, people do not write a poem at one stroke. Instead, they start with an initial draft and go through several revisions to produce the final poem. Hrešková et al. [Hrešková and Machová, 2017] proposed an interactive genetic algorithm-based Haiku generation system. Users score each population of Haiku according to their subjective preferences, and the best Haiku in accordance with the user’s preference is obtained after multiple iterations. Yang et al. [Yang and Hagiwara, 2016] developed an interactive genetic algorithm-based Waka generation system which retrieved literary pieces related to the content and emotion of the user-provided texts from custom databases. After each generation, the system calculated the fitness of Waka using automatic and manual evaluation mechanisms, and produced the next population through a series of genetic operations.

With the rise of deep learning, poetry generation systems can better understand the semantic meaning of poetry through neural networks. Wu et al. [Wu et al., 2017] studied Haiku generation with various RNN models and SeqGAN, and compared the perplexity values obtained from these models. Shao et al. [Shao et al., 2018] used the LSTM language model to generate Haiku with specified keywords. Kaga et al. [Kaga et al., 2017] and Konishi et al. [Konishi et al., 2017] investigated how to generate Haiku preferred by the general public. Kaga et al. [Kaga et al., 2017] trained a LSTM language model with Haiku created by professional scholars, and then fine-tuned the model using Haiku created by non-professional scholars to make the generated poems easier to understand. Konishi et al. [Konishi et al., 2017] provided the generator and the discriminator of SeqGAN with different datasets, so that the discriminator could judge the generation with the values of the general public. Hirota et al. [Hirota et al., 2018] adopted a method similar to that of Konishi et al. [Konishi et al., 2017], aiming to avoid plagiarism when generating Haiku by Neural Probabilistic Language Model (NPLM). Masada et al. [Masada and Takasu, 2018] trained a GRU language model with Waka poems segmented into bigrams, and scored the generated results using a LDA-based method. Their proposed scoring method could select generated poems with a wider variety of subsequences than those selected by RNN output probabilities.

In this paper, we conduct the study of Waka generation based on Transformer and/or VAE. On the one hand, neural networks are used to model the semantic relations in Waka, which could improve the model’s understanding of the semantic meaning. On the other hand, we consider both the objective and subjective methods to evaluate Waka poems generated by our models in a comprehensive way.

### 3 Our approach

This section describes the method we used. We first formalize our task. Suppose  $V$  denotes the vocabulary and  $c \in V$  denotes a keyword specified by the user, our goal is to generate a Waka poem  $x = (x_1, x_2, \dots, x_T)$ ,  $x_i \in V$  of length  $T$ , which contains the given keyword  $c$ . According to the morae pattern,  $x$  should be sequentially divisible into 5 phrases with fixed morae counts. That is to say, there exists exactly 4 subscripts  $i, j, k, l$  subject to  $1 \leq i < j < k < l < T$ , making the following statements true:

$$\sum_{t=1}^i s(x_t) = \sum_{t=j+1}^k s(x_t) = 5 \quad (1)$$

$$\sum_{t=i+1}^j s(x_t) = \sum_{t=k+1}^l s(x_t) = \sum_{t=l+1}^T s(x_t) = 7 \quad (2)$$

where  $s(w)$  denotes the morae count of a given word  $w \in V$ . As Fig. 1 illustrates, a common separator Sep exists between the adjacent phrases, with the morae count defined as zero, i.e.,  $s(\text{Sep}) = 0$ .

Next, we introduce two Waka generation models based on the Transformer. The first model (TVAE) contains a single latent variable, while the second model (WakaVT) contains a sequence of latent variables.

### 3.1 TVAE

Recently, the Transformer model and its variants [Tay et al., 2020] have proven fruitful for a variety of NLP tasks. The original Transformer model [Vaswani et al., 2017] is an encoder-decoder framework designed for the sequence-to-sequence task. Each encoder layer consists of two sub-layers - the multi-head self-attention sublayer and the position-wise feed-forward network. Compared with the encoder layer, each decoder layer inserts an encoder-decoder attention sublayer between the above two sublayers to perform multi-head attention over the output of the encoder stack. Another difference between the encoder and the decoder is that the attention mechanisms of the former are non-causal, while those of the latter are causal, which is achieved through attention masks.

Self-attention mechanism is the core part of Transformer. In general, the attention mechanism learns the alignment between queries and keys through a compatibility function. For self-attention, a softmax function is applied to the output of the dot product between the queries and keys to obtain the alignment scores, which are then used to calculate the weighted sum of the values as output. This process is formulated as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (3)$$

where  $Q$ ,  $K$ , and  $V$  are the matrices of queries, keys, and values, respectively, and  $d_k$  is the column count of  $K$ . The multi-head self-attention mechanism is then formulated as:

$$\text{MultiHead}(X) = \text{concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h) W^O \quad (4)$$

$$\text{head}_i = \text{Attention}\left(XW_i^Q, XW_i^K, XW_i^V\right) \quad (5)$$

where  $X$  is the matrix of the input sequence, and  $W_i^Q, W_i^K, W_i^V, W^O$  are trainable parameters. In the self-attention mechanism, the dot product between the queries and keys drives the self-alignment process, where each token in the sequence learns to gather information from each other [Tay et al., 2020]. On top of that, multi-head self-attention allows the model to focus on the information from different representation subspaces [Vaswani et al., 2017]. As we can see from the calculation, the self-attention mechanism fairly aligns any two positions in the input sequence (regardless of the distance between them), and thus can easily capture long-term dependencies.

#### 3.1.1 Model architecture

Based on the standard Transformer architecture, we construct the naive CVAE model (named TVAE) as a baseline. The overall structure of TVAE is shown in Fig. 2. We use precisely the same structure as the encoder layer of the original Transformer model on the encoder side. We leak the future information of the input sequence to the recognition network through the non-causal attention mechanism of the encoder layers. To encode the input sequence into a fixed-length vector, we follow the BERT model’s approach [Devlin et al., 2019], which adds a <CLS> token at the beginning of the input sequence and takes the output of the corresponding position as the encoded representation of the entire input. Both the recognition network and the prior network are Multilayer Perceptrons (MLPs) used to map inputs into the latent space. For the recognition network, the inputs consist of the output of the encoder and the embedding of the keyword. As a comparison, the prior network only takes the embedding of the keyword as input. The reparameterization trick [Kingma and Welling, 2014] is used here to solve the problem of non-differentiable sampling process of latent variables.

On the decoder side, we remove the encoder-decoder attention sublayer. We directly add the latent variable together with the embedding vector of <SOS> token, and then input the summation into the decoder. Inspired by the Ctrl model [Keskar et al., 2019], we put the keyword at the beginning of the decoder input sequence as a control code to ensure that the generated poem contains this keyword. Thanks to the properties of the self-attention mechanism, the path length of any position in the sequence to the keyword is 1, thus enabling the keyword to have direct control over the content of the whole poem. Finally, the decoder generates a poem based on the latent variable and the input keyword. Note that the decoder layers contain causal attention since this allows the decoder to generate texts in an autoregressive manner.

Let  $\theta$  be the parameters of the prior network and the decoder, and  $\phi_r$  be the parameters of the encoder and the recognition network. Like the original CVAE model [Sohn et al., 2015], the loss function of TVAE is defined as:

$$\begin{aligned} L_{CVAE} &= -\mathbb{E}_{q_{\phi_r}(z|x,c)} \left[ \log \frac{p_{\theta}(x, z | c)}{q_{\phi_r}(z | x, c)} \right] \\ &= -\mathbb{E}_{q_{\phi_r}(z|x,c)} [\log p_{\theta}(x | z, c)] + D_{KL}(q_{\phi_r}(z | x, c) || p_{\theta}(z | c)) \end{aligned} \quad (6)$$

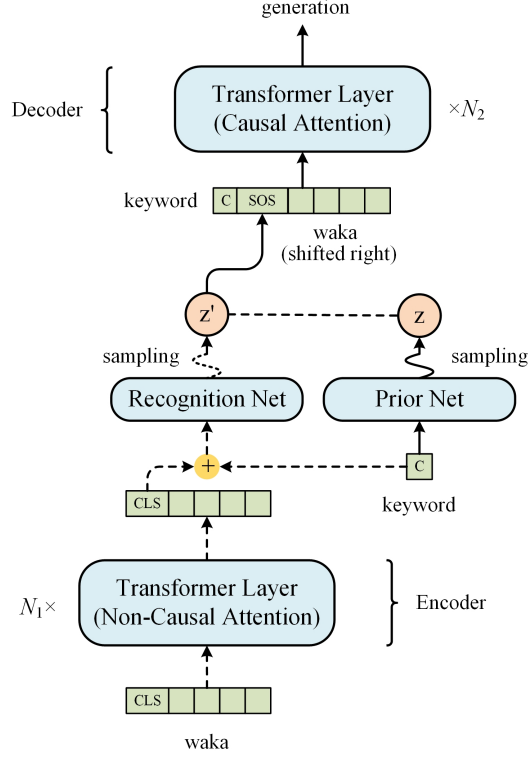


Figure 2: Architecture of TVAE. Dotted lines represent the connection that only appears in the training phase, and “+” represents the concatenation operation.

In this formula,  $D_{KL}$  denotes the Kullback-Leibler divergence between the posterior and prior. We use the bag-of-word auxiliary loss [Zhao et al., 2017] and the KL cost annealing technique [Bowman et al., 2016] to alleviate the posterior collapse problem. The learning objective is finalized as:

$$L = L_{CVAE} + \alpha L_{BOW} \quad (7)$$

where  $L_{BOW}$  denotes the bag-of-word auxiliary loss and  $\alpha$  denotes a balancing coefficient.

### 3.1.2 Morae constraint satisfaction using additive masks

Different words have different number of moraes, but the overall morae pattern of Waka is fixed, being 5, 7, 5, 7, 7. We design a mask-based method to ensure the generated poems follow the correct morae pattern. Let a 5-morae phrase or a 7-morae phrase in a Waka contain  $n$  words, and the morae pattern of the phrase is expressed as a sequence  $(a_1, a_2, \dots, a_n)$ , in which  $a_i$  denotes the morae count of the  $i$ -th word in the phrase. The total morae count  $m = \sum_{i=1}^n a_i$  should satisfy  $m = 5$  or  $m = 7$ , respectively. We define a sequence  $(l_1, l_2, \dots, l_n)$ , in which  $l_i = m - \sum_{k=1}^{i-1} a_k$  denotes the upper bound of the morae count of the  $i$ -th word predicted by the decoder. Since the morae count of the  $i$ -th word must not exceed  $l_i$ , we can mask out all words with the morae count larger than  $l_i$  when the decoder predicts the  $i$ -th word. Let the size of the dictionary be  $D$ , the morae count of the  $j$ -th word  $w_j$  in the dictionary be  $s(w_j)$ , and the logit and the predicted probability of the  $i$ -th word be  $o_i$  and  $p_i = \text{softmax}(o_i)$ , respectively, then the mask  $m_i = (m_{i,1}, m_{i,2}, \dots, m_{i,D})$  at the current time step is defined as:

$$m_{i,j} = \begin{cases} 0, & \text{if } s(w_j) \leq l_i \\ -\infty, & \text{if } s(w_j) > l_i \end{cases} \quad (8)$$

The output is then changed to  $p'_i = \text{softmax}(o_i + m_i)$ , where  $p'_i$  satisfies that for any  $s(w_j) > l_i$ , there is  $p'_{i,j} = 0$ . This mechanism masks out all words that may cause the morae count to exceed the limit when generating each word<sup>4</sup>.

<sup>4</sup>Note that additive masks can also be used to avoid incomplete moraes. However, we have found that most generation could follow the correct pattern even if we don't consider this problem.

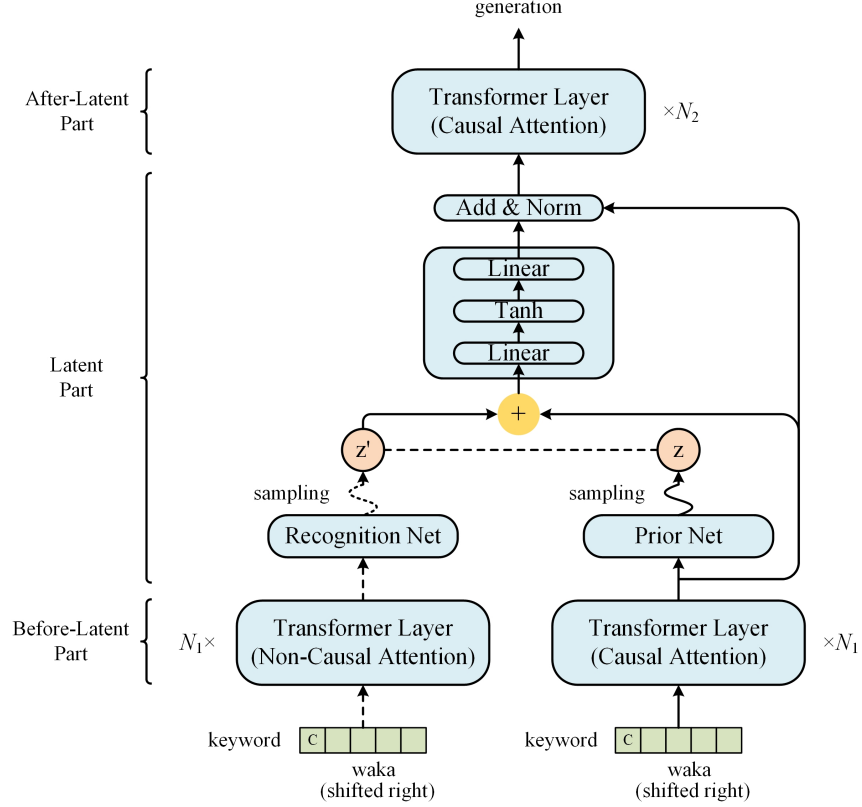


Figure 3: The architecture of WakaVT. Dashed lines represent connections that occur only during the training phase, and “+” represents the concatenation operation.

In the training phase, additive masks can be calculated during data processing to speed up the training process. In the inference phase, they can be incrementally calculated. After generating a new word, the additive mask of the next word to be generated is calculated. Note that all special tokens (except for the generic unknown word token <UNK>) are specified with zero morae. For simplicity, the morae count of <UNK> is specified as an integer larger than 7 to ensure that no out-of-vocabulary (OOV) words appear in the generated results.

### 3.2 WakaVT

TVAE uses a single latent variable that obeys Gaussian distribution, which has limited ability to model the diversity of the input data. In contrast, using a sequence of latent variables for text modeling can improve the diversity of the generated samples [Du et al., 2018, Lin et al., 2020, Schulz et al., 2018]. Inspired by this, we propose WakaVT, a Waka generation model based on the sequential variational Transformer. A sequence of latent variables is introduced to capture the variability of each position into a respective uni-modal distribution. In addition, WakaVT uses the same methods for keyword and morae constraint satisfaction as TVAE.

#### 3.2.1 Model architecture

The structure of WakaVT, as shown in Fig. 3, can be divided into three parts: the before-latent part, the latent part, and the after-latent part.

The before-latent part follows tightly after the input layer and is used to encode the input sequence and pass the observed information to the latent part. As in Fig. 3, there are 2 separate stacks of Transformer layers in this part. On the left side, a stack of  $N_1$  layers adopts non-causal attention mechanism to allow the positions to attend to each other, which is similar to the encoder of TVAE. On the right side, another stack of  $N_1$  layers contains causal attention to prevent the positions from attending to the subsequent positions. Similar to the decoder of TVAE, this allows the model to predict each token depending only on the known outputs when generating a text.



The latent part is the core of WakaVT, comprising the recognition network, the prior network, and the fusion network. The recognition network and the prior network are MLPs, which parameterize the conditional distributions of the corresponding latent variables  $z'$  and  $z$ . Assuming that the inputs of the recognition network and the prior network are  $o^{(r)}$  and  $o^{(p)}$ , respectively, the posterior and prior at time step  $t$  are approximated by:

$$q_{\phi_r}(z'_t | x_{1:T}, c) = \mathcal{N}(\mu'_t, \sigma'^2_t) \quad (9)$$

$$[\mu'_t, \log \sigma'^2_t] = \text{MLP}(o^{(r)}_t) \quad (10)$$

$$q_{\phi_p}(z_t | x_{1:t-1}, c) = \mathcal{N}(\mu_t, \sigma^2_t) \quad (11)$$

$$[\mu_t, \log \sigma^2_t] = \text{MLP}(o^{(p)}_t) \quad (12)$$

The fusion network is among the recognition network, the prior network, and the after-latent part. It's used to integrate the latent information represented by  $z'$  (in the training phase) or  $z$  (in the inference phase), and the prior observed information represented by  $o^{(p)}$ , into a single representation. Notably,  $z'$  (or  $z$ ) and  $o^{(p)}$  are in different forms. The former is a randomly observed point in the latent space, while the latter is a deterministic text representation encoded by Transformer layers. Therefore, we introduce a fusion mechanism to merge them. The fusion process is calculated as:

$$m_t = V \tanh(Wz'_t + Uo^{(p)}_t) \quad (13)$$

where  $V$ ,  $W$  and  $U$  are trainable parameters. Afterwards, residual connection and residual dropout [Vaswani et al., 2017] are applied to make it easier for optimization. The final output of the latent-part is obtained as:

$$o^{(m)}_t = \text{LayerNorm}(o^{(p)}_t + \text{Dropout}(m_t)) \quad (14)$$

The after-latent part consists of a stack of  $N_2$  Transformer layers and an output layer (not shown in Fig. 3). Its functionality is to predict the probability distribution of each token according to the latent information from the latent-part, the prior observed information from the before-latent part, and the input keyword from the user. During training, the decoding process of WakaVT is formulated as:

$$p_\theta(x | z', c) = \prod_t p_\theta(x_t | x_{1:t-1}, z'_{1:t}, c) \quad (15)$$

where  $\theta$  denotes trainable parameters of all parts of the model. During inference, since the recognition network is removed, we perform the decoding process using  $z_{1:t}$  instead of  $z'_{1:t}$ .

### 3.2.2 Fused Multilevel Self Attention Mechanism

Waka has a natural hierarchical structure of four levels as shown in Fig. 1. There exist semantic connections and transitions among the phrases and the sentences. We take the Waka in Fig. 1 as an example<sup>5</sup>. The words むなしき(hollow), そら(sky), and かぜ(wind) in Kami-no-Ku are all about scenery and in close relation to each other. Similarly, the words ひと(people) and わすれ(forget) in Shimo-no-Ku are used to express the poet's inner feelings and are also well correlated. However, there is a content transition from the scenery description to the psychological description, and the relations between the words in Kami-no-Ku and those in Shimo-no-Ku would not be so clear if we ignored the context. Therefore, semantic relations at different hierarchical levels may vary a lot in a Waka poem, and the relationship among the words more distant from each other is more difficult to understand.

The original self-attention mechanism directly models the relationship between any two positions in the input sequence, unaware of the natural hierarchical feature of Waka. If such feature can be fully utilized, it will be easier for the model to understand the semantic relations at different levels. Therefore, we present the Fused Multilevel Self Attention (FMSA), which gathers features at different levels through different multi-head attention mechanisms and integrates them through a fusion mechanism as in Fig. 4. We incorporate FMSA into all Transformer layers of WakaVT.

FMSA consists of three multi-head attention sublayers and a fusion unit. The global multi-head attention sublayer is applied to capture the global dependencies in Waka, as is the standard multi-head attention mechanism, connecting

<sup>5</sup>Selected from the collection of *Bunpo Hyakushu*, edited by *Emperor Go-Uta*.

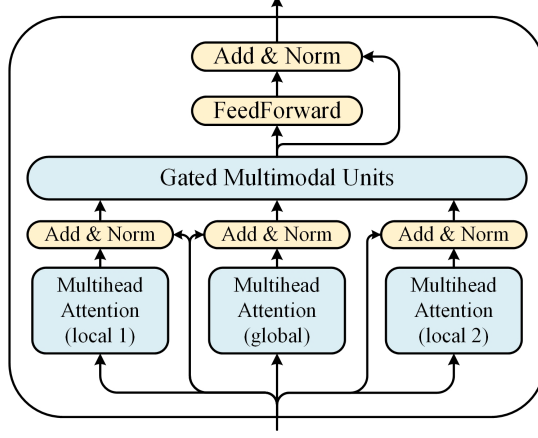


Figure 4: FMSA-based Transformer Layer

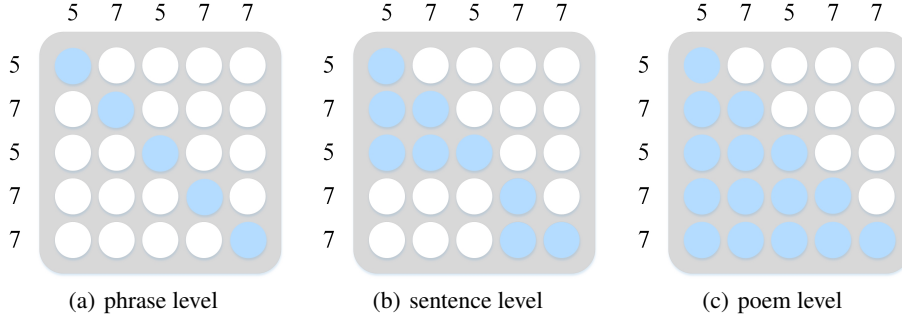


Figure 5: Attention Masks

all pairs of the positions in the input sequence. Two local multi-head attention sublayers are introduced for the 5/7-morae phrase pattern and the sentence structure of Kami-no-Ku and Shimo-no-Ku, respectively. They restrict the self-alignment process in a local range to specially attend to the relations within phrases and sentences. Local attention mechanisms are implemented through attention masks. Fig. 5 illustrates the format of the causal attention mask of each attention sublayer.

Supposing that after the residual connection and the layer normalization, the outputs of the three attention sublayers are  $o^{f_1}, o^{f_2}, o^{f_3}$ , respectively (in no specific order), the fusion process is fulfilled through Gated Multimodal Unit (GMU) [Ovalle et al., 2017], which is calculated as:

$$h_t^{f_i} = \tanh \left( W_{f_i} o_t^{f_i} \right) \quad (16)$$

$$z_t^{f_i} = \text{sigmoid} \left( W_{z^{f_i}} \left[ o_t^{f_1}, o_t^{f_2}, o_t^{f_3} \right] \right) \quad (17)$$

$$o_t^{\hat{f}} = \sum_{i=1}^3 z_t^{f_i} * h_t^{f_i} \quad (18)$$

$$o^{\hat{f}} = \left( o_1^{\hat{f}}, o_2^{\hat{f}}, \dots, o_T^{\hat{f}} \right) \quad (19)$$

with  $W_{f_i}$  and  $W_{z^{f_i}}$  as trainable parameters and  $o^{\hat{f}}$  as the output of GMU.

GMU first transforms the inputs from different modalities into feature vectors through nonlinear transformations, and then controls the contribution of the calculated features to the overall output through gating strategies using sigmoid function. Intuitively, GMU guides FMSA to explore how to integrate the most essential features from different hierarchical levels into a single vector. In this way, the Transformer layers with FMSA can achieve fine-grained representations of hierarchical features of the input sequence.

### 3.2.3 Learning

Given a keyword  $c$  and a sample  $x = x_{1:T}$  of length  $T$ , the loss function of WakaVT can be written as:

$$\begin{aligned} L_{CVAE} &= -\mathbb{E}_{q_{\phi_r}(z_{1:T}|x_{1:T},c)} \left[ \sum_t \log \frac{p_{\theta}(x_t | x_{1:t-1}, z_{1:t}, c) p_{\theta}(z_t | x_{1:t-1}, c)}{q_{\phi_r}(z_t | x_{1:T}, c)} \right] \\ &= -\sum_t \mathbb{E}_{q_{\phi_r}(z_{1:t}|x_{1:T},c)} [\log p_{\theta}(x_t | x_{1:t-1}, z_{1:t}, c)] \\ &\quad + \sum_t D_{KL}(q_{\phi_r}(z_t | x_{1:T}, c) \| q_{\phi_p}(z_t | x_{1:t-1}, c)) \end{aligned} \quad (20)$$

where  $D_{KL}$  is the Kullback-Leibler divergence between the posterior and prior.

To alleviate the posterior collapse problem, we introduce the SBOW auxiliary loss [Du et al., 2018], which forces the latent variables to capture more useful information from the input sequence. It is defined as:

$$p_{\xi}(x_{t:t+l-1} | z_t, x_{1:t-1}, c) = \text{MLP}(z_t, o_t^{(p)}) \quad (21)$$

$$L_{SBOW} = -\sum_t \mathbb{E}_{q_{\phi_r}(z_t|x_{1:T},c)} [\log p_{\xi}(x_{t:t+l-1} | x_{1:t-1}, z_t, c)] \quad (22)$$

where  $l$  is the truncated length, which simply means that only the continuous  $l$  tokens started from  $x_t$  are considered when making the calculation at current time step  $t$ . Ultimately, the total loss of WakaVT is the weighted sum of  $L_{CVAE}$  and  $L_{SBOW}$ :

$$L = L_{CVAE} + \alpha L_{SBOW} \quad (23)$$

## 4 Experiments

### 4.1 Data

We build a large and available Waka dataset containing anthologies from the 8th to 16th centuries AD, including *Manyoshu*, *Nijuichidaishu*, *Uta-awase* and a variety of private collections. All data are collected from the *International Research Center for Japanese Studies* database<sup>6</sup>. After data cleansing, we obtained a total of 171801 Waka poems, each of which is written in one continuous line using historical kana orthography. For the sake of availability, the voiced sounds need to be recognized and marked with dakuten diacritic marks, and the kana characters should be segmented into words to build the vocabulary. Therefore, we first train a sequence labeling model based on Conditional Random Fields (CRF) [Sutton and McCallum, 2012] using 20 thousand Waka poems with properly labeled voiced sounds, and then use it to automatically annotate our dataset. Next, we applied Web茶まめ<sup>7</sup> (*Web Chamame*), a morphological analysis tool built by the *National Institute for Japanese Language and Linguistics*, to perform word segmentation. TextRank algorithm [Yang et al., 2018a] is adopted to extract a keyword for each Waka. Finally, 10,000 poems are randomly selected from the dataset as the validation set, 5,000 as the test set, and the remaining 156,801 as the training set.

### 4.2 Baselines

For Waka generation with given keywords, we design and implement three baselines as below. To ensure the generated Waka strictly conforms to the morae constraint, the additive mask-based method proposed in Sect. 3.1.2 is applied to each model.

**RNN-VAD:** A GRU-based conditional variational autoregressive decoder. A sequence of latent variables subject to a multimodal distribution is used to model the high variability of texts. It is adapted from the model proposed in [Du et al., 2018]. We retained the forward RNN and the backward RNN, removed the encoder and the associated attention module, and added a linear layer for each RNN to convert the embedding vector of the input keyword into the initial hidden states.

**TLM:** A Transformer-based conditional language model. Texts with controllable content can be generated using various control codes [Keskar et al., 2019].

<sup>6</sup><http://db.nichibun.ac.jp/pc1/ja/category/waka.html>

<sup>7</sup><https://chamame.ninjal.ac.jp/>

**TVAE:** A Transformer-based CVAE model, as discussed in Sect. 3.1. Unlike WakaVT, TVAE models data using the unimodal distribution of a single latent variable.

FMSA is proposed based on the self-attention mechanism and is independent of specific model architectures. We applied FMSA to TLM, TVAE and WakaVT to investigate its effect on each model.

### 4.3 Model settings

The 128-dimensional word embeddings pretrained by Fasttext [Bojanowski et al., 2017] were adopted to initialize each model’s embedding layer, with a dictionary size of 6649. For all Transformer-based models, the multi-head attention sublayer was made up of 4 heads and the feedforward network contained an inner layer with 512 units. The TLM model consisted of a stack of 4 Transformer layers, and the same setting was adopted for the counterparts of TVAE and WakaVT. Specifically, for TVAE, the corresponding parameters  $N_1, N_2$  (see Fig. 2) were set as  $N_1 = N_2 = 4$ , and for WakaVT (Fig. 3),  $N_1 = N_2 = 2$ . Both the forward and the backward GRU of RNN-VAD contained 3 layers with a hidden size of 256, and layer normalization was applied to each GRU cell as in [Ba et al., 2016]. The size of the latent variables was set to 128 everywhere. In the training stage, the models were all trained by Adam optimizer [Kingma and Ba, 2015], with a learning rate of 0.0001 and a mini-batch size of 32. Linear scheduling of KLD loss was applied to the training process of all VAE models, combined with appropriate auxiliary loss to alleviate the posterior collapse. Specifically, BOW loss was adopted for TVAE, while SBOW loss with a truncated length of 5 was adopted for RNN-VAD and WakaVT. In the inference stage, we used beam search algorithm with the beam width set to 20 for all models.

### 4.4 Evaluation design

It is notoriously difficult to judge the quality of poems generated by computers. We conduct both objective and subjective evaluation to comprehensively evaluate the poems generated by each model.

#### 4.4.1 Objective evaluation

We employ three evaluation metrics in our experiments. They are defined as follows:

**PPL & KLD:** We calculated the reconstruction perplexity (PPL) and the Kullback-Leibler divergence (KLD) between the posterior and prior for each model on the test set. A well-trained model should achieve a low PPL value and a small but non-trivial KLD value.

**Novelty:** The novelty is aimed to evaluate to what extent the generated Waka differs from the training set: whether the model simply copies the pieces from the training set or generates new pieces itself. Given a training set  $C$  and a generation set  $S$ ,  $\text{Nov}_w$  is defined as:

$$\text{Nov}_w(S) = \frac{1}{|S|} \sum_{i=1}^{|S|} \text{Nov}_w(S_i) \quad (24)$$

$$\text{Nov}_w(S_i) = 1 - \max_{j=1}^{|C|} \{\text{Dice}(S_i, C_j)\} \quad (25)$$

with  $S_i$  as a poem in  $S$ ,  $C_j$  as a poem in  $C$ , and  $\text{Dice}(S_i, C_j)$  as the Sørensen-Dice coefficient<sup>8</sup> between  $S_i$  and  $C_j$ . Intuitively,  $\text{Nov}_w$  only measures the novelty at the word level, thus we define a metric at the phrase level to make the evaluation more comprehensive. Given a set of Waka  $A$ ,  $\text{Phr}_5(A)$  and  $\text{Phr}_7(A)$  as the set of the corresponding 5-morae phrases and 7-morae phrases, respectively,  $\text{Nov}_{s,n}$  is defined as:

$$\text{Nov}_{s,n}(S) = \frac{|\text{Phr}_n(S) - \text{Phr}_n(C)|}{|\text{Phr}_n(S)|}, n = 5 \text{ or } 7 \quad (26)$$

We use 1000 keywords to generate 1000 Waka poems (one for each keyword) for each model to calculate the above metrics.

**Diversity:** The diversity is designed to measure how different the generated poems are with different specified inputs. A low diversity metric signals that the model always generates similar results and lacks creative potential. Similar to the novelty metrics, we define a metric at the word level as  $\text{Div}_w$  and another at the phrase level as  $\text{Div}_{s,n}$ :

$$\text{Div}_w(S) = \frac{1}{|S|} \sum_{i=1}^{|S|} \text{Div}_w(S_i) \quad (27)$$

<sup>8</sup>[https://en.wikipedia.org/wiki/S%C3%B8rensen%E2%80%93Dice\\_coefficient](https://en.wikipedia.org/wiki/S%C3%B8rensen%E2%80%93Dice_coefficient)

$$\text{Div}_w(S_i) = 1 - \max_{j=1}^{|S|, j \neq i} \{\text{Dice}(S_i, S_j)\} \quad (28)$$

$$\text{Div}_{s,n}(S) = \frac{|\text{Phr}_n(S)|}{\sum_{p \in \text{Phr}_n(S)} \text{Count}_S(p)}, n = 5 \text{ or } 7 \quad (29)$$

where  $\text{Count}_S(p)$  denotes the number of times the phrase  $p$  appears in  $S$ . The novelty metric and diversity metric are calculated using the same generation set  $S$ .

#### 4.4.2 Subjective evaluation

We subjectively evaluate the linguistic quality of Waka from three aspects: fluency, coherence, and meaningfulness. The details are shown in Table 1. In the experiment, it was found that the additive mask-based method proposed in Sect. 3.1.2 was sufficient to ensure that most generated poems met the morae constraint<sup>9</sup>. Therefore, we do not design a particular evaluation method for the correctness of formats. For the sake of fairness, the 100 most frequent keywords were selected, and each model generated a poem for each keyword. We invited 3 experts major in classical Japanese poetry to conduct a blind review of the generated poems on a scale from 1 (very poor) to 5 (very good). The scores of the poems are averaged as the final score for each model.

Table 1: Subjective evaluation criteria

Metric	Detail
Fluency	Whether a poem is grammatically and syntactically satisfied.
Coherence	Whether a poem is semantically coherent among the phrases and between Kami-no-ku and Shimo-no-ku.
Meaningfulness	Whether a poem conveys an artistic conception with an optional use of figures of speech, such as kakekotoba, engo, and makurakotoba.
Overall	Mean value of the above three metrics.

#### 4.5 Quantitative analysis

Table 2: Objective evaluation results.  $\downarrow$  indicates the lower the better, while  $\uparrow$  indicates the higher the better.

Model	PPL $\downarrow$	KLD $\uparrow$	Novelty $\uparrow$			Diversity $\uparrow$		
			Nov $_w$	Nov $_{s,5}$	Nov $_{s,7}$	Div $_w$	Div $_{s,5}$	Div $_{s,7}$
TLM	15.14	-	0.3909	0.0661	0.2104	0.2979	0.4326	0.5110
TLM+FMSA	14.21	-	0.3776	0.0249	0.1153	0.2999	0.3801	0.4683
TVAE	12.72	10.71	0.4161	0.0767	0.3675	0.4394	0.3651	0.5903
TVAE+FMSA	12.09	8.66	0.4027	0.0351	0.2258	0.4128	0.3287	0.5237
RNN-VAD	8.09	18.86	0.4201	<b>0.1792</b>	0.4962	0.4883	0.6055	0.8243
WakaVT	6.98	20.18	0.4273	0.1482	0.4542	0.4978	<b>0.6171</b>	0.7966
WakaVT+FMSA	<b>5.60</b>	<b>26.70</b>	<b>0.4400</b>	0.1446	<b>0.4966</b>	<b>0.5182</b>	0.6085	<b>0.8310</b>

The objective evaluation results are shown in Table 2. We can find that models with a sequence of latent variables (WakaVT, WakaVT+FMSA, and RNN-VAD) achieved smaller PPL values and larger KLD values than those with a single latent variable (TVAE and TVAE+FMSA). Additionally, TVAE and TVAE+FMSA got smaller PPL values than the models without latent variables (TLM and TLM+FMSA). This indicates that a single latent variable can improve the model’s capability to reconstruct the input sequence, and a sequence of latent variables may further boost the results by capturing more detailed semantic information. Compared with WakaVT and RNN-VAD, WakaVT+FMSA achieved the lowest PPL value and the highest KLD value, due to the high modeling capability of the Transformer architecture along with the auxiliary effect of the FMSA module.

It’s observed that models with a sequence of latent variables outperformed other models in terms of Nov $_w$ , Nov $_{s,5}$ , and Nov $_{s,7}$ , which confirms that latent variables can indeed improve the novelty of words and phrases. Compared

<sup>9</sup> Among the generated poems used for objective evaluation, the proportion of morae constrained poems generated by TLM, TVAE, WakaVT, and RNN-VAD were 92.7%, 98.2%, 99.5%, and 100%, respectively.

with RNN-VAD, WakaVT+FMSA achieved larger  $\text{Nov}_w$ , smaller  $\text{Nov}_{s,5}$ , and similar  $\text{Nov}_{s,7}$ . This demonstrates that WakaVT+FMSA has comparable capability to RNN-VAD in terms of novelty.

Through the Pearson Correlation Analysis, we find that  $\text{Nov}_w$  and  $\text{Div}_w$  show a strong positive correlation ( $r = 0.96$ ). The result still holds for the correlation between  $\text{Nov}_{s,5}$  and  $\text{Div}_{s,5}$  ( $r = 0.94$ ), and that between  $\text{Nov}_{s,7}$  and  $\text{Div}_{s,7}$  ( $r = 0.96$ ). Similar to the conclusion of the novelty evaluation, models with a sequence of latent variables outperformed other models in terms of  $\text{Div}_w$ ,  $\text{Div}_{s,5}$ , and  $\text{Div}_{s,7}$ . Among them, WakaVT+FMSA scored the highest in  $\text{Div}_w$  and  $\text{Div}_{s,7}$  and WakaVT scored the highest in  $\text{Div}_{s,5}$ . This proves that a sequence of latent variables can improve the diversity at both the word level and the phrase level. Additionally, the combination of the Transformer and FMSA can further boost the results.

Table 3: Subjective evaluation results

Model	Fluency	Coherence	Meaningfulness	Overall
TLM	4.30	3.66	2.93	3.63
TLM+FMSA	4.49	3.98	3.49	3.99
TVAE	4.36	3.69	3.16	3.74
TVAE+FMSA	<b>4.57</b>	3.94	3.55	4.02
RNN-VAD	4.38	3.83	3.37	3.86
WakaVT	4.36	3.84	3.38	3.86
WakaVT+FMSA	4.52	<b>4.17</b>	<b>3.74</b>	<b>4.14</b>

Table 3 shows the results of the subjective evaluation. For each model, the scores of different metrics could be ranked in a decreasing order as fluency, coherence, meaningfulness. This indicates that for any model, it is relatively easy to form smooth sentences, rather difficult to ensure semantic coherence, and the greatest challenge is to express desired artistic conception. Significantly, models with FMSA module (TLM+FMSA, TVAE+FMSA, WakaVT+FMSA) obtained higher scores of all metrics than the other models, confirming the effect of FMSA in improving the linguistic quality of the generated poems. Among the models with FMSA module, WakaVT+FMSA scored the highest while TLM+FMSA and TVAE +FMSA obtained similar scores. Specifically, TVAE+FMSA scored the highest in fluency, while WakaVT+FMSA scored the second-best with small difference from TVAE+FMSA. In terms of coherence and meaningfulness, TLM+FMSA and TVAE+FMSA scored similarly, while WakaVT+FMSA had a significant advantage over them ( $p < 0.05$ )<sup>10</sup>.

Overall, RNN-VAD performed similar to WakaVT+FMSA in terms of novelty and diversity, yet worse than WakaVT+FMSA in terms of linguistic quality. TLM+FMSA and TVAE+FMSA were not as good at linguistic quality as WakaVT+FMSA, and were even significantly inferior to WakaVT+FMSA in novelty and diversity. Therefore, we conclude that WakaVT+FMSA outperformed baseline models significantly in the overall quality of generated poems, including linguistic quality, novelty and diversity.

## 4.6 Qualitative analysis

### 4.6.1 FMSA visualizations

To analyze the mechanism of FMSA more intuitively, we input a Waka poem from the test set (same as the example discussed in Sect. 3.2.2) into WakaVT and WakaVT+FMSA, and conduct a visualization analysis of the alignment weights. The calculated PPL values of WakaVT and WakaVT+FMSA with this poem as input are 5.39 and 3.15, respectively. We visualized the alignment weights at the first layer of the after-latent part (averaging the weights of all 4 heads), with results shown in Fig. 6.

The visualization results of the self-attention mechanisms at different levels are shown in Fig. 6(a), Fig. 6(b), and Fig. 6(c), respectively. It can be seen that FMSA local attention (Fig. 6(a)) at the phrase level only aligns the tokens within 5/7-morae phrases. For example, in the second phrase, the local attention attends to the dependency between むなしき(hollow) and そら(sky). FMSA local attention at the sentence level (Fig. 6(b)) aligns the tokens within Kami-no-ku and Shimo-no-ku. We can see that the model pays extra attention to the lexical dependencies among different phrases in a sentence, and the dependencies within the phrases are weakened (the color becomes lighter). For example, the alignment weight between むなしき(hollow) and そら(sky) has decreased, meanwhile the model puts a special emphasis on the connection between かぜ(wind) and the above two words. The FMSA global attention (Fig. 6(c)) aligns all the words in the Waka. It is observed that much attention is paid to adjacent words, but the dependencies

<sup>10</sup>Unilateral T-test is used for significance test.

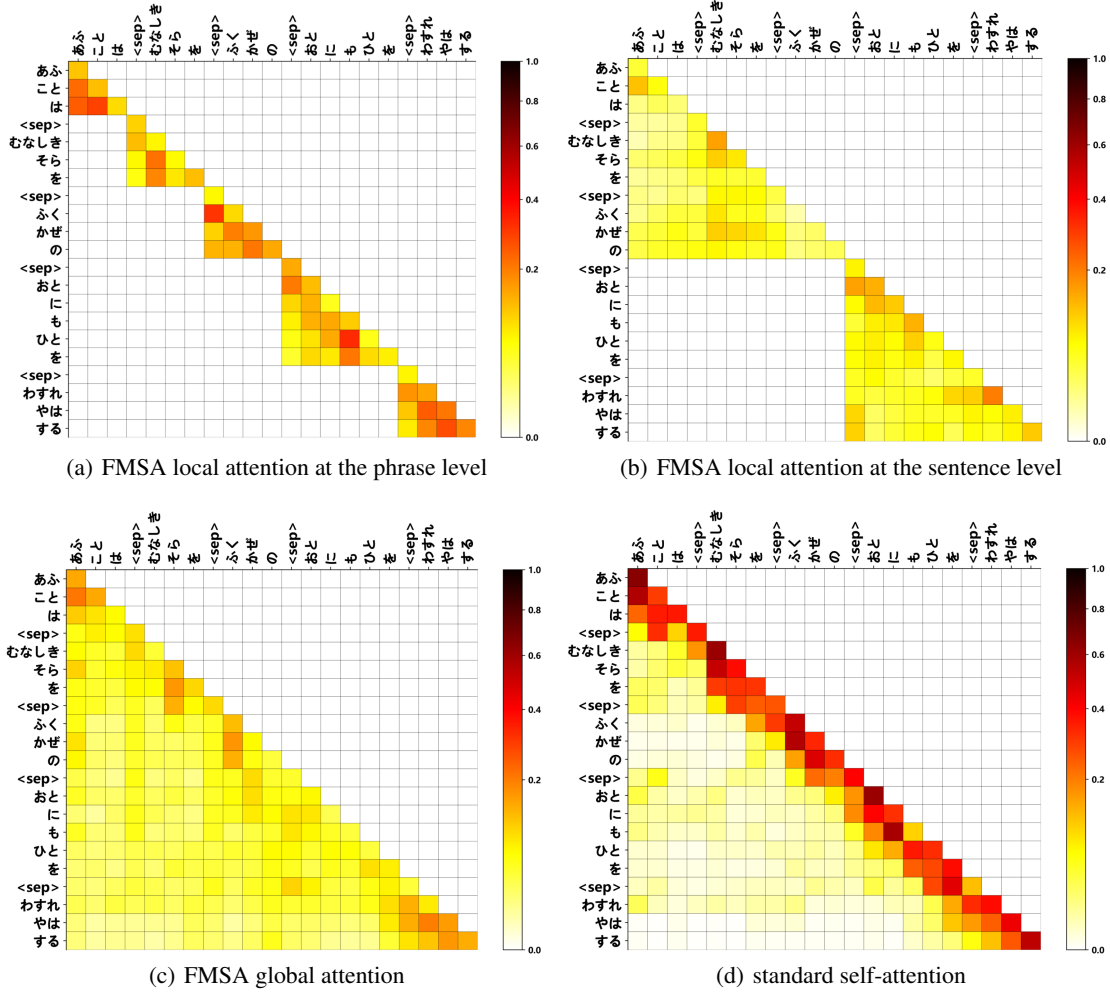


Figure 6: FMSA visualization results

within Kami-no-ku and Shimo-no-ku, as well as those between the sentences, are also taken into consideration. For example, the connections among あふ(meeting), そら(sky), and かぜ(wind) in the Kami-no-ku (the wind in the sky is a metaphor for lovers meeting), and the connection between かぜ(wind) in the Kami-no-ku and おと(sound) in the Shimo-no-ku are marked in deeper colors. The visualization results of the standard self-attention are shown in Fig. 6(d). Compared with FMSA global attention, it pays little attention to the connections among the words located far from each other, but too much attention to local lexical connections.

The following observations are drawn from the above results. Firstly, the standard self-attention mechanism models the lexical connections in Waka from only a single view, while FMSA is able to do it from 3 views through different self-attention modules. As a result, FMSA can deal with the lexical connections at different hierarchical levels, effectively leveraging hierarchical features of Waka. Secondly, the standard self-attention mechanism may tend to model local lexical connections at the phrase level, ignoring certain important connections in a wider range, which is more difficult to capture. On the contrary, for FMSA, the phrase-level dependency modeling is completed by the corresponding local attention sublayer, leaving the other two attention sublayers to capture longer dependencies.

#### 4.6.2 Case study

In order to better illustrate the impact of latent variables on the generated poems, we input the keyword はる(spring) into TLM+FMSA, TVAE+FMSA, RNN-VAD, and WakaVT+FMSA, and then compare the generated results. Table 4 shows the generation of each model.

Table 4: Generated results of each model with keyword はる(spring). Each poem is splitted into two lines, i.e. Kami-no-ku and Shimo-no-ku. Repeated words/phrases are underlined.

Model	Generated Waka
TLM+FMSA	はるのきる—かすみのころも—うすければ かすみのころも—ほころびにけり Spring comes in rosy cloud dress. Too thin the cloth is, the dress splits.
TVAE+FMSA	はるのよの—ありあけのつき—かげもなし なほありあけの—ありあけのそら Spring night, in the dawn sky, waning moon loses the light. Silent dawn as ever, the silent dawn sky.
RNN-VAD	うめのはな—ちりくるほとは—なけれども はるをのこして—すぎぬべらなり Wintersweet, blooming as always. Turn around, left Spring behind.
WakaVT+FMSA	みよしのの—やまほととぎす—ながきよの やまのみやこの—はるをまつかな Wait, Cuckoos on Yoshinoyama, following the lingering night, is awaked capital in full spring.

The Waka generated by the model without latent variables, TLM+FMSA, has actually borrowed the first two phrases from the Waka created by human beings<sup>11</sup>. However, the fourth phrase is a repetition of the second phrase, thus the poem is all about rosy clouds from the very beginning to the end, lacking the content development. Although a single latent variable is used in TVAE+FMSA, the content is still relatively simple. The word ありあけ(dawn) was used three times in the generated Waka, destroying the content’s integrity. In contrast, the Waka generated by the models with a sequence of latent variables, including RNN-VAD and WakaVT+FMSA, successfully avoided meaningless repetition of words and phrases. By associating はる(spring) with うめのはな(the plum blossom, which is to be withered), RNN-VAD depicts a picture of early spring. WakaVT+FMSA associates the word はる(spring) with ほととぎす(cuckoos), やま(mountain), and みよこ(capital). The description of the cuckoos waiting for the arrival of spring in the capital on Yoshino Mountain diversifies the intended meaning of はる(spring). Moreover, it’s creative to combine やま(mountain), representing the beauty of the nature, and みよこ(capital), representing power and prosperity, in a single phrase. It can be seen that a sequence of latent variables can optimize the model to generate Waka with diversified words, rich content and novel ideas. In contrast, models with a single latent variable or without latent variables may cause word repetition, resulting in meaningless content.

More samples generated by WakaVT+FMSA are shown in Table 5. We invited 3 experts to evaluate each poem individually. Their comments confirmed that WakaVT+FMSA could indeed generate ingenious and innovative Waka poems.

## 5 Conclusion and future work

In this paper, we present a novel Waka generation model, WakaVT, which combines advantages of the latent variable model and the self-attention mechanism. Specifically, a sequence of latent variables is incorporated to model word-level variability in Waka data. Moreover, the proposed self-attention based mechanism FMSA is adopted to learn the hierarchical feature of Waka. Experimental results show that a sequence of latent variables significantly improves the novelty and diversity of the generated poems, and FMSA effectively promotes the linguistic quality in terms of fluency, coherence, and meaningfulness. Considering the objective and subjective evaluation results, our model has apparent advantages over the baselines. In future works, we will apply pretraining methods with large-scale classical literary datasets to make the model better learn the semantic meanings of ancient texts, which we believe can further improve the linguistic quality of generated Waka.

<sup>11</sup>In the training set, at least 5 poems created by humans start with はるのきる—かすみのころも.



Table 5: Samples generated by WakaVT+FMSA and the corresponding comments on them. The keyword used to generate each poem is underlined.

	<p>あけてゆくーみねのこのはのーこずゑよりーはるかにつづくーさをしかの<u>こゑ</u>  The morning light climbs the treetops of the top mountain,  and the <u>cry</u> of stags spreads far and wide.</p>
<p>reviewer 1 reviewer 2 reviewer 3</p>	<p>The scenery is <i>fresh with a wild imagination</i>, and the connection of several images is ingenious.  This poem is <i>thought-provoking</i> as it gradually transitions from visual to auditory. The far-reaching cry of the stag can further arouse the longing for the loved one. It is indeed a classical Waka poem.  This Waka is <i>beautiful as well as touching</i> under the background of autumn. The author expresses parting sadness through the auditory perspective of the stag's courtship sound.</p>
	<p>こひしさは一ひとのこころにーさよふけてーわかなみだこそーおもひしらるれ  Night approaches, my love is longing in his heart faraway,  my <u>tears</u> come and see by all.</p>
<p>reviewer 1</p>	<p>The expression is <i>proficient and sophisticated with strong feeling</i>. There is a jump in content between the first two phrases and the last three phrases.</p>
<p>reviewer 2</p>	<p>This poem is <i>grammatically coherent and accurate</i>, which reminds us of the primal 「恋しさに思ひみだれてねぬる夜の深き夢ちをうつうともかな」, a sleepless night troubled by love.</p>
<p>reviewer 3</p>	<p>This poem is <i>simple in language and sincere in emotion</i>. The poet depicts the pain of lovesickness and reveals his infinite sadness for incomplete love.</p>
	<p>ふるさとのーあとをたづねてーなつくさのーしげみにかかるーをのの<u>かよひ</u>ち  I trace the road of my hometown,  the lush grass of summer covers the return <u>path</u> in the wilderness.</p>
<p>reviewer 1</p>	<p>The poet's longing for home is reflected. It is quite <i>interesting</i> to know how the fourth phrase connects the third and fifth phrases.</p>
<p>reviewer 2</p>	<p>Lush summer grass covered the trace of the hometown road. The entire poem is <i>closely connected and understandable in meaning</i>. The accurate grasp of the image of summer grass is quite touching.</p>
<p>reviewer 3</p>	<p>This poem is composed with plain language, yet it conveys authentic emotion. It is <i>simple in style, beautiful in artistic conception, and fluent in language</i>. The poet truly expresses profound nostalgia.</p>
	<p>はてもなきーまがきのくさはーおく<u>つゆ</u>にーおもひあまりてーやどるつきかな  Endless weeds by the fence are covered with <u>dew</u>,  yearning was reflected on the dew by moonlight.</p>
<p>reviewer 1</p>	<p>The idea of connecting certain images such as weeds, dew, and moon is quite <i>intriguing</i>. The fourth phrase is <i>affectionate</i>.</p>
<p>reviewer 2</p>	<p>The weeds beside the fence correspond to melancholy, and dew corresponds to the moon. This poem is <i>quite classical and interesting</i>, and it feels like <i>Kokinshu</i>.</p>
<p>reviewer 3</p>	<p>This Waka is <i>beautiful in artistic conception</i>. The dew and the moon's mutual reflection presents an ethereal and tranquil scene, revealing the poet's romantic feelings of nature.</p>

---

## References

- Hugo Gonalo Oliveira. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 11–20, 2017.
- Shulei Ji, Jing Luo, and Xinyu Yang. A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions. *arXiv:2011.06801*, 2020.
- Simon Colton, Jacob Goodwin, and Tony Veale. Full-face poetry generation. In *Proceedings of the Third International Conference on Computational Creativity (ICCC)*, pages 95–102, 2012.
- Hugo Gonalo Oliveira. Poetryme: a versatile platform for poetry generation. *Computational Creativity, Concept Invention, and General Intelligence (C3GI)*, 1:1–21, 2012.
- Ruli Manurung, Graeme Ritchie, and Henry Thompson. Using genetic algorithms to create meaningful poetic text. *Journal of Experimental & Theoretical Artificial Intelligence*, 24(1):43–64, 2012.
- Jing He, Ming Zhou, and Long Jiang. Generating chinese classical poems with statistical machine translation models. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 1650–1656, 2012.
- Rui Yan, Han Jiang, Mirella Lapata, Shou-De Lin, Xueqiang Lv, and Xiaoming Li. I, poet: automatic chinese poetry composition through a generative summarization framework under constrained optimization. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2197–2203, 2013.
- Tim Van de Cruys. Automatic poetry generation from prosaic text. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2471–2480, 2020.
- Zhipeng Guo, Xiaoyuan Yi, Maosong Sun, Wenhao Li, Cheng Yang, Jiannan Liang, Huimin Chen, Yuhui Zhang, and Ruoyu Li. Jiuge: A human-machine collaborative chinese classical poetry generation system. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 25–30, 2019.
- Rafal Rzepka and Kenji Araki. Haiku generator that reads blogs and illustrates them with sounds and images. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2496–2502, 2015.
- Naoko Tosa, Hideto Obara, and Michihiko Minoh. Hitch haiku: An interactive supporting system for composing haiku poem. In *Proceedings of the 7th International Conference on Entertainment Computing*, pages 209–216, 2008.
- Atsushi Hirota, Natsuki Oka, Masahiro Araki, and Kazuaki Tanaka. Haiku generation by seggan that divided learning data set. In *Proceedings of the Twenty-fourth Annual Meeting of the Association for Natural Language Processing*, pages 1292–1295, 2018.
- Xianchao Wu, Momo Klyen, Kazushige Ito, and Zhan Chen. Haiku generation using deep neural networks. In *Annual Conference of the Language Processing Society*, pages 1133–1136, 2017.
- Ming Yang and Masafumi Hagiwara. A text-based automatic waka generation system using kansei. *International Journal of Affective Engineering*, 15(2):125–134, 2016.
- Tomonari Masada and Atsuhiko Takasu. Lda-based scoring of sequences generated by rnn for automatic tanka composition. In *International Conference on Computational Science*, pages 395–402, 2018.
- Jiyuan Zhang, Yang Feng, Dong Wang, Yang Wang, Andrew Abel, Shiyue Zhang, and Andi Zhang. Flexible and creative chinese poetry generation using neural memory. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1364–1373, 2017.
- Juntao Li, Yan Song, Haisong Zhang, Dongmin Chen, Shuming Shi, Dongyan Zhao, and Rui Yan. Generating classical chinese poems via conditional variational autoencoder and adversarial training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3890–3900, 2018.
- Lei Shen, Xiaoyu Guo, and Meng Chen. Compose like humans: Jointly improving the coherence and novelty for modern chinese poetry generation. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020.
- Jiachen Du, Wenjie Li, Yulan He, Ruifeng Xu, Lidong Bing, and Xuan Wang. Variational autoregressive decoder for neural response generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3154–3163, 2018.
- Zhaojiang Lin, Genta Indra Winata, Peng Xu, Zihan Liu, and Pascale Fung. Variational transformers for diverse response generation. *arXiv:2003.12738*, 2020.

- 
- Philip Schulz, Wilker Aziz, and Trevor Cohn. A stochastic decoder for neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1243–1252, 2018.
- Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *arXiv:1909.05858*, 2019.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, 2017.
- Liming Deng, Jie Wang, Hangming Liang, Hui Chen, Zhiqiang Xie, Bojin Zhuang, Shaojun Wang, and Jing Xiao. An iterative polishing framework based on quality aware masked language model for chinese poetry generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7643–7650, 2020.
- Xingxing Zhang and Mirella Lapata. Chinese poetry generation with recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 670–680, 2014.
- Marjan Ghazvininejad, Xing Shi, Yejin Choi, and Kevin Knight. Generating topical poetry. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1183–1191, 2016.
- Daisy Zhe Wang, Wei He, Hua Wu, Haiyang Wu, Wei Li, Haifeng Wang, and Enhong Chen. Chinese poetry generation with planning based neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1051–1060, 2016.
- Xiaopeng Yang, Xiaowen Lin, Shunda Suo, and Ming Li. Generating thematic chinese poetry using conditional variational autoencoders with hybrid decoders. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4539–4545, 2018a.
- Cheng Yang, Maosong Sun, Xiaoyuan Yi, and Wenhao Li. Stylistic chinese poetry generation via unsupervised style disentanglement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3960–3969, 2018b.
- Xiaoyuan Yi, Ruoyu Li, Cheng Yang, Wenhao Li, and Maosong Sun. Mixpoet: Diverse poetry generation via learning controllable mixed latent space. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9450–9457, 2020.
- Joanna Misztal and Bipin Indurkha. Poetry generation system with an emotional personality. In *Proceedings of the Fifth International Conference on Computational Creativity (ICCC)*, pages 72–81, 2014.
- Huimin Chen, Xiaoyuan Yi, Maosong Sun, Wenhao Li, Cheng Yang, and Zhipeng Guo. Sentiment-controllable chinese poetry generation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4925–4931, 2019.
- Zhiqiang Liu, Zuohui Fu, Jie Cao, Gerard de Melo, Yik-Cheung Tam, Cheng Niu, and Jie Zhou. Rhetorically controlled encoder-decoder for modern chinese poetry generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1992–2001, 2019.
- Jinyi Hu and Maosong Sun. Generating major types of chinese classical poetry in a uniformed framework. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4658–4663, 2020.
- Piji Li, Haisong Zhang, Xiaojiang Liu, and Shuming Shi. Rigid formats controlled text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 742–751, 2020.
- Dayiheng Liu, Quan Guo, Wubo Li, and Jiancheng Lv. A multi-modal chinese poetry generation model. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2018.
- Linli Xu, Liang Jiang, Chuan Qin, Zhe Wang, and Dongfang Du. How images inspire poems: Generating classical chinese poetry from images with memory networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5618–5625, 2018.
- Yael Netzer, David Gabay, Yoav Goldberg, and Michael Elhadad. Gaiku: Generating haiku with word associations norms. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, pages 32–39, 2009.
- Tsan Martin Wong, On Kee Sigma Lee, and Hon Wai Andy Chun. Automatic haiku generation using vsm. In *Proceedings of the 7th WSEAS International Conference on Applied Computer and Applied Computation Science (ACACOS’08)*, pages 318–323, 2008.
- Takuya Ito, Jumpei Ono, and Takashi Ogata. Haiku generation using gap techniques. In *Proceedings of the 2018 International Conference on Artificial Intelligence and Virtual Reality*, pages 93–96, 2018.
- Miroslava Hrešková and Kristína Machová. Haiku poetry generation using interactive evolution vs. poem models. *Acta Electrotechnica et Informatica*, 17(1):10–16, 2017.

- 
- Guanming Shao, Yosuke Kobayashi, and Jay Kishigami. Traditional japanese haiku generator using rnn language model. In *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)*, pages 263–264, 2018.
- Yuta Kaga, Ryo Iehara, Atsushi Hirota, Ryosuke Kanajiri, Yukihiro Wakigami, Bungo Konishi, Seigo Matsuo, Chie Fukada, Kazuaki Tanaka, and Natsuki Oka. Learning method of neural probabilistic language model for preferable haiku generation. In *Information Processing Society of Japan (IPSJ) Kansai-Branch Convention*, pages 1–5, 2017.
- Bungo Konishi, Atsushi Hirota, Seigo Matsuo, Ryo Iehara, Soichiro Obara, Yuta Kaga, Joji Tsuruda, Yukihiro Wakigami, Ryosuke Kanajiri, Chie Fukada, Kazuaki Tanaka, and Natsuki Oka. Generation of haiku preferable for ordinary people by seqgan. In *Information Processing Society of Japan (IPSJ) Kansai-Branch Convention*, pages 1–3, 2017.
- Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *arXiv:2009.06732*, 2020.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.
- Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, pages 1–14, 2014.
- Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pages 3483–3491, 2015.
- Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664, 2017.
- Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, 2016.
- John Edison Arevalo Ovalle, Tamar Solorio, Manuel Montes-y-Gómez, and Fabio A. González. Gated multimodal units for information fusion. In *International Conference on Learning Representations*, pages 1–17, 2017.
- Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *Found Trends Mach Learn*, 4(4):267–373, 2012.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv:1607.06450*, 2016.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, pages 1–15, 2015.