A Study of Bidirectional and Attention-Enhanced RNN, GRU, and LSTM in Poetry Generation

1st Fengcheng Yu
School of Intelligent Engineering, Sun Yat-sen University
ShenZhen, China
yufch3@mail2.sysu.edu.cn

***Abstract*—This study expands on previous work by exploring the application of character-based recurrent neural networks (CharRNN) in poetry generation, focusing on Long Short-Term Memory (LSTM), basic Recurrent Neural Networks (RNN), and Gated Recurrent Units (GRU). We introduced attention mechanisms and bidirectional structures to enhance the coherence and complexity of the generated texts. Utilizing a diverse dataset, including Chinese poetry, English articles, Chinese song lyrics, Japanese articles, and Linux script languages, we applied all model variants specifically to the Chinese poetry dataset to assess their performance comprehensively. Evaluation was conducted using the BERT-CCPoem system and manual scoring from Tsinghua University's AI Research Institute. Results showed that while advanced features like attention mechanisms increased training costs and did not significantly enhance the quality of poetry, simpler LSTM and GRU models performed better and were more cost-effective. The study also examined the models' capabilities with multilingual and diverse datasets, highlighting the potential and challenges of using deep learning in artistic creation. Future research directions include optimizing model structures and exploring new text generation technologies, aiming to overcome limitations in generating long texts and maintaining semantic coherence.**

***Index Terms*—Chinese poetry generation, CharRNN, LSTM, GRU, attention mechanism, bidirectional network, natural language processing**

I. Introduction

As one of the essences of Chinese culture, Chinese classical poetry has been full of profound artistic and emotional value for thousands of years. Its unique rhythm, format, and rich cultural connotations make it an important resource for studying Chinese literature and linguistics. However, with the acceleration of modernization, the creation skills and appreciation ability of traditional poetry have gradually been marginalized, requiring a lot of literary cultivation and historical knowledge, which is a big challenge for contemporary society, especially the younger generation.

Faced with this challenge, deep learning technology provides new possibilities for the creation and inheritance of classical poetry. By simulating the thinking and creative style of ancient poets, machine learning models, especially recurrent neural networks (RNN[16]), long short-term memory networks (LSTM[9]), and gated recurrent units (GRU[4]), have shown their potential in automatically generating poetry. These technologies can not only learn and simulate the language structure and rhythmic features of ancient poetry but also reproduce the emotions[20] and artistic conception of poetry to a certain extent.[22] [13]

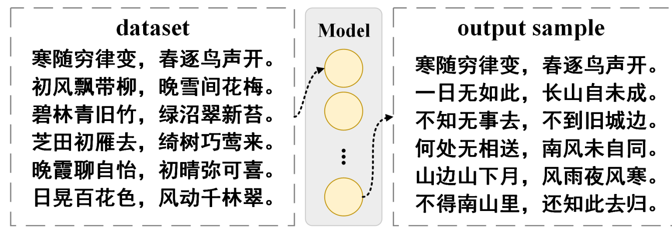

Fig. 1. Example output of the poetry generation model

Although existing machine learning models can generate poems of a certain quality, there is still a clear gap in artistry and literary depth compared with the works of human poets.[20] [22] [13] In order to improve the quality of generated poetry and better simulate the creative process of human poets, this study introduces attention mechanisms and bidirectional network structures to enhance the model's ability to capture context and the deep meaning of poetry. Through a systematic evaluation[19] of these advanced model structures, we aim to explore their practicality and efficiency in classical poetry generation.

In addition, this study also focuses on the adaptability and versatility of the model. By testing on a variety of datasets, including texts in different languages and formats, we evaluate the performance of LSTM, RNN, and GRU in different cultural and linguistic environments. These experiments not only test the generation ability of the model, but also ensure the literary value and artistry of the generated works through a complex quality assessment system - combining automatic evaluation tools and manual scoring.

Through in-depth analysis and evaluation, this study reveals the potential and limitations of machine learning technology in classical poetry creation, and proposes improvement directions and new research perspectives. This not only helps promote the application of artificial intelligence in the field of literary creation, but also provides scientific and technological support for the inheritance and innovation of classical poetry.

The main contributions of this paper are as follows:

1) The performance and effect of LSTM, RNN and GRU in poetry generation are systematically evaluated.

2) The performance and results of LSTM, RNN and GRU in poetry generation after adding Attention mechanism[2] and changing to bidirectional[17] network are systematically evaluated.
3) For the first time, the actual application effect of attention mechanism and bidirectional network configuration in poetry generation task is compared and analyzed.
4) A comprehensive evaluation of the quality of poetry generation based on different models is provided, including the results of automatic evaluation tools and manual scoring.

As shown in table I, the abbreviations and meanings of the symbols used in this article are as follows. The code for this project will be open-source soon: https://github.com/ffengc/NLP-PoetryModels.

TABLE I
NOTATION USED IN THE PAPER

Symbol	Meaning
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory Model
GRU	Gated Recurrent Unit Model
bi/Bi-*	Abbreviation for Bidirectional Model

II. RELATED WORK

Automatic poetry generation is an important research direction in the field of natural language processing. This section will review in detail several key technologies and their applications in this field.[7]

A. Multimodal poetry generation

Traditional poetry generation is mainly based on text, while multimodal[11] poetry generation methods attempt to combine images and text, thus opening up a new way to create automatic poetry. This type of method mainly uses image recognition technology to extract poetic clues[23] [12] in the image (such as natural landscapes, emotional atmosphere, etc.), and uses these clues as the basis for generating poetry. In this process, multi-adversarial training and policy gradient methods are used to optimize the generation model to ensure that the generated text is closely related to the image content and poetic. To this end, the researchers built a deep joint vision-poetry embedding system, through which the relationship between objects, emotions, and scenes in the image and poetry can be learned simultaneously. [10]

B. Poetry generation based on LSTM and GRU

LSTM and GRU are effective tools for processing long-term dependent information in text sequences[8], especially in the field of poetry generation, they can maintain the coherence and style consistency of the text. Studies have shown that models based on these recurrent neural networks can not only process standard Chinese or English poetry, but also adapt to poetry generation tasks in multiple languages such as Arabic and Urdu[1] [14]. By comparing different configurations, GRU shows better performance in maintaining the correctness of poetry sentences and vocabulary, showing adaptability and efficiency in diverse language environments. [25] [30] [26]

C. Application of Attention Mechanism

In poetry generation, the application of attention mechanisms, especially multi-head attention mechanisms, greatly improves the model's ability to handle complex text structures. By using a Transformer-XL-based architecture, the model can more accurately simulate and learn the deep semantic relationships between Chinese characters when generating text. This advanced attention model helps generate more semantically rich and coherent poetry by enhancing the connections between characters, while also improving the artistic beauty and literary value of the poetry. [29] [21] [27]

D. Style transfer and sentiment classification

Style transfer technology allows the poetry generation model to imitate the style of a specific author[24] [5] [28], thereby generating poems with the characteristics of a specific author. This not only increases the diversity of generated poems, but also improves their artistry. Sentiment classification technology further enables the generated poems to express richer emotions[6] [3], such as joy, sadness, or love. The combination of these technologies not only improves the emotional expressiveness of the poetry text, but also enhances the emotional resonance of the readers. [15] [18]

Through in-depth exploration of the above research, this paper expands on the CharRNN model, introduces the attention mechanism and bidirectional network structure, and develops multiple model configurations to explore the impact of different technology combinations on the quality of poetry generation. This study not only focuses on the generation of Chinese poetry, but also attempts to apply the model to multilingual texts such as English articles, Chinese lyrics, Japanese articles, and Linux scripts, in order to discover a wider range of application potential and deeply understand the performance and advantages of various model configurations in different text generation tasks.

III. METHOD

This study explores poetry generation based on recurrent neural networks, especially the application of LSTM, RNN and GRU, and enhances the performance of the model by introducing attention mechanism and bidirectional network structure. The following is a detailed introduction to the operation mechanism of each model and its implementation in this study. The overall structure of the model is shown in Figure 2.

A. Basic recurrent neural network module

1) Long Short-Term Memory (LSTM): Module (B) in fig. 2 shows the structural details of LSTM. LSTM is specially designed to solve the gradient vanishing problem of traditional RNN when processing long sequence data. LSTM optimizes

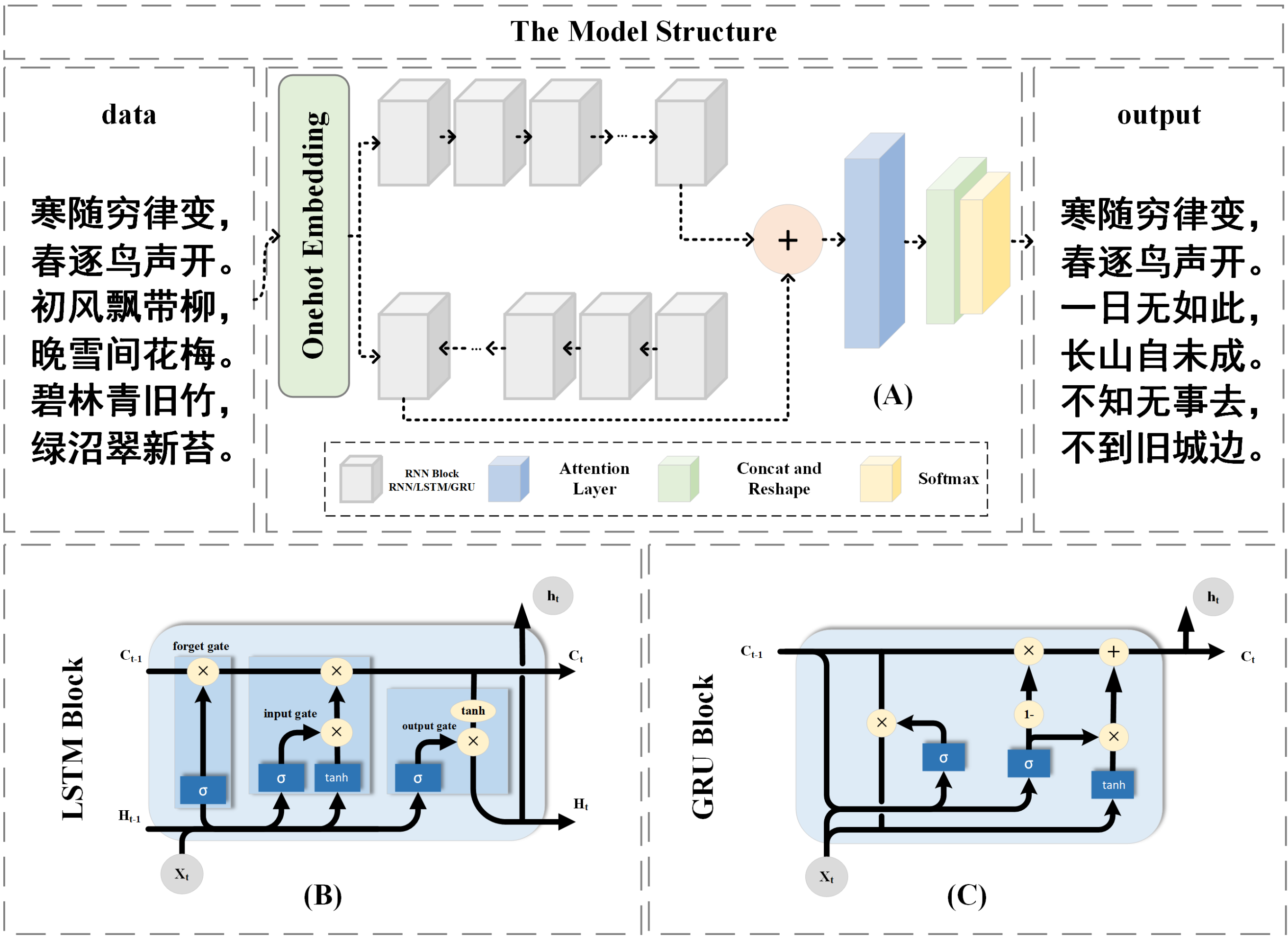

Fig. 2. Our Model

the information flow by introducing a complex gating mechanism, including input gate, forget gate and output gate. The role of each gate is described in detail by eq. (1).

$$
\begin{aligned}
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) && (a) \\
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) && (b) \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) && (c) \\
c_t &= f_t \cdot c_{t-1} + i_t \cdot \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) && (d) \\
h_t &= o_t \cdot \tanh(c_t) && (e)
\end{aligned}
\tag{1}
$$

Among in eq. (1):

(a) Input gate: responsible for deciding what new information is allowed to enter the cell state at the current time step.
(b) Forget gate: controls what information in the previous cell state is retained.
(c) Output gate: determines what information is passed from the cell state to the output.
(d) Cell state: is the core of LSTM, ensuring that the network can retain information for a long time.
(e) Output: represents the network output at the current time step, which will be used in the next time step or used to generate the final poem.

i_t, f_t, o_t are the activation values of the input gate, forget gate, and output gate at time step t. W_i, W_f, W_o, W_c are the corresponding weight matrices. (b_i, b_f, b_o, b_c) are the corresponding bias terms. h_{t-1} is the output of the previous time step. x_t is the input of the current time step. (c_{t-1}) is the cell state of the previous time step. c_t is the cell state of the current time step. σ is the Sigmoid activation function, which is used to control the opening and closing of the gate. $\tanh$ is the hyperbolic tangent activation function, which is used to compress data in the range of $([-1, 1])$.

2) Basic Recurrent Neural Network (RNN): RNN uses recursive connections in the hidden layer so that the output from the previous time step can be used as part of the input of the current time step, which is suitable for processing sequence data. Its basic hidden state update formula is shown in eq. (2).

$$
h_t = \tanh(W_h \cdot [h_{t-1}, x_t] + b_h) \tag{2}
$$

In a basic recurrent neural network (RNN), h_t is the hidden state of the current time step, and h_{t-1} is the hidden state of the previous time step. x_t is the input of the current time step. W_h is the weight matrix of the hidden layer, and b_h is the corresponding bias term. The state update uses the $\tanh$ function, which is a hyperbolic tangent activation function to compress the data in the range $[-1, 1]$.

3) Gated Recurrent Unit (GRU): As shown in the module (C) in fig. 2 shows the design of GRU. GRU is a further simplification of LSTM. It simplifies the complexity of the model by merging the input gate and the forget gate into an update gate and adding a reset gate, while maintaining the ability to capture long-distance dependency information, as shown in eq. (3).

$$\begin{aligned} z_t &= \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) & (a) \\ r_t &= \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) & (b) \\ \tilde{h}_t &= \tanh(W \cdot [r_t \cdot h_{t-1}, x_t] + b) & (c) \\ h_t &= (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t & (d) \end{aligned} \tag{3}$$

(a) Update gate: determines how much of the old state is retained.
(b) Reset gate: determines how much past information to ignore when computing new candidate hidden states.
(c) Candidate hidden state: Provides a possible new state.
(d) Final hidden state: Combine the old state and the new candidate state to generate the current state.

In a Gated Recurrent Unit (GRU), z_t and r_t are the activations of the update gate and reset gate, respectively. W_z and W_r are the weight matrices of the corresponding gates, and b_z and b_r are the bias terms. h_{t-1} is the hidden state of the previous time step. x_t is the input of the current time step. $\tilde{h}_t$ is the candidate hidden state, processed by the $\tanh$ function. The final hidden state h_t is generated by combining the old state and the new candidate state, using the σ function to control the behavior of the update and reset gates.

B. Improvement Mechanism

1) Attention Mechanism: The attention mechanism used in this model allows the network to focus on specific parts of the input sequence when generating the words or characters of each poem. We use the output of the bidirectional LSTM to calculate the alignment scores at different time steps, which are converted into attention weights after being processed by the softmax function. This process can be described by 4.

$$\begin{aligned} a_t &= v^T \tanh(W_a h_t + b_a) & (a) \\ \alpha_t &= \text{softmax}(a_t) & (b) \\ c_t &= \textstyle\sum_{t'} \alpha_{t'} h_{t'} & (c) \end{aligned} \tag{4}$$

- Alignment score: (1) indicates the calculation of the similarity between the current hidden state and all input states.
- Attention weight: (2) indicates that the alignment score is processed by the softmax function and converted into a standardized attention weight.
- Context vector: (3) indicates that the relevant context of the current word/character is calculated by weighted average hidden state, using the attention weight as the weight.

a_t is the alignment score, which is obtained by calculating the similarity between the current hidden state and all input states; v^T is the weight vector in the attention mechanism, which is used for transposition and multiplication with other vectors; $\tanh$ is the hyperbolic tangent activation function, which compresses the data into the range of $[-1, 1]$ to help stabilize numerical calculations; W_a is the weight matrix used to calculate the alignment score; h_t is the hidden state of the current time step; b_a is the bias term for alignment score calculation; α_t is the attention weight, which is calculated by normalizing the alignment score with the softmax function to ensure that the sum of all weights is 1 to maintain the validity of the distribution; c_t is the context vector, which represents the weighted average hidden state, reflecting the relevant context of the current word or character; $\sum$ represents the summation operation, which is used to summarize the weighted hidden states of all time steps; t' is the index of different time steps, which is used to traverse the hidden states of the entire sequence during the summation process.

2) Bidirectional recurrent network: In the bidirectional LSTM structure, information is not only transmitted from front to back, but also fed back from back to front. This design enables the model to consider both the previous and the following information at the same time, which is the key to understanding and generating poems with complex structures and semantics. The bidirectional structure provides a comprehensive contextual perspective by processing forward and reverse sequences in parallel.

C. Implementation details

In terms of specific implementation, our model is built based on the TensorFlow framework. The model input layer uses high-dimensional word embedding technology, and each word is converted into a high-dimensional space vector. These vectors are further processed through the bidirectional LSTM layer. In addition, in order to adapt to the rhythm and style of different poems, we introduced a dynamically adjusted attention mechanism, which not only improves the flexibility of the model, but also enhances the beauty and expressiveness of the poems.

During the training phase, we used a variety of optimization techniques including but not limited to the Adam optimizer to ensure rapid convergence of the model. To prevent gradient explosion, gradient clipping technology is introduced to limit the maximum allowed value of the gradient by setting a threshold. In addition, dropout technology is used to prevent overfitting and improve the model's generalization ability on novel samples.

The entire model training process is conducted under strict control, and each step is designed to optimize the quality of poetry generation so that it reaches high artistic and technical standards. Through careful loss function design, the model is

Fig. 3. Example of Datas

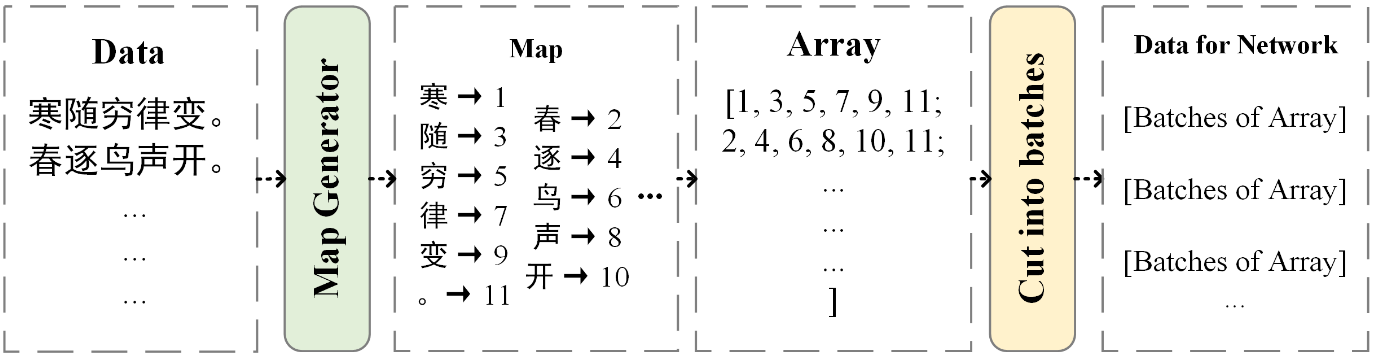

Fig. 4. Data Preprocessing Process

able to accurately evaluate and optimize the difference between generated verses and target verses, thereby performing well in diverse and complex poetry generation tasks.

IV. Experiment

A. Data collection

In this project, I prepared five datasets as shown below.

- Chinese Poetry: Collected from professional poetry databases and literary websites, ensuring coverage of various styles and genres from ancient to modern times, a total of 72,514 lines of five-character poems.
- Chinese lyrics: Jay Chou's lyrics were captured from multiple popular music platforms to ensure text diversity, totaling 6016 lines.
- English articles: Shakespeare articles extracted from online news portals and e-books, totaling 127,604 lines.
- Japanese paragraphs: taken from Japanese news websites and literary works, ensuring the modernity of the language and the breadth of the culture, a total of 1,691 lines.
- The C language code of the Linux kernel, a total of 241,465 lines.

The sources of the five data sets are all from open source data on Github. The data examples are shown in fig. 3.

B. Data Preprocessing

In this project, data preprocessing is divided into 6 steps, and the specific functions and descriptions are as follows.

1) Encoding and reading: Text files are read using UTF-8 encoding. UTF-8 encoding supports the correct parsing of multiple characters including English letters and Chinese characters, ensuring the compatibility and correctness of multilingual texts.
2) Character Mapping Construction: Create a mapping (dictionary) from each unique character to a unique index. This mapping relationship is to convert text from string form to a numerical form that the model can process. This mapping is the basis for the model to understand text, because the neural network cannot understand text characters directly, but learns by processing numerical data.
3) Vocabulary Limitation: Based on the preset maximum vocabulary size (e.g. 3500 different characters), if the number of characters in the text exceeds this number, the characters will be frequency analyzed and only the most frequently occurring characters will be retained. This not only helps reduce the complexity of the model, but also improves processing speed and efficiency.
4) Text digitization: Using the character-to-index mapping dictionary constructed above, the entire text is converted into an array of digital indices. This array is exactly the input data for neural network training.
5) Data serialization and batch processing: According to the needs of neural network training, the digitized text data is divided into multiple small batches, each batch contains multiple sequences of fixed length. Each sequence consists of a specified number of consecutive character indexes. Such serialization processing enables the model to learn the dependencies between characters during training.
6) Save processors and mappings: Save the character-to-index mapper (TextConverter object) to disk for reuse during training and generation, ensuring that character

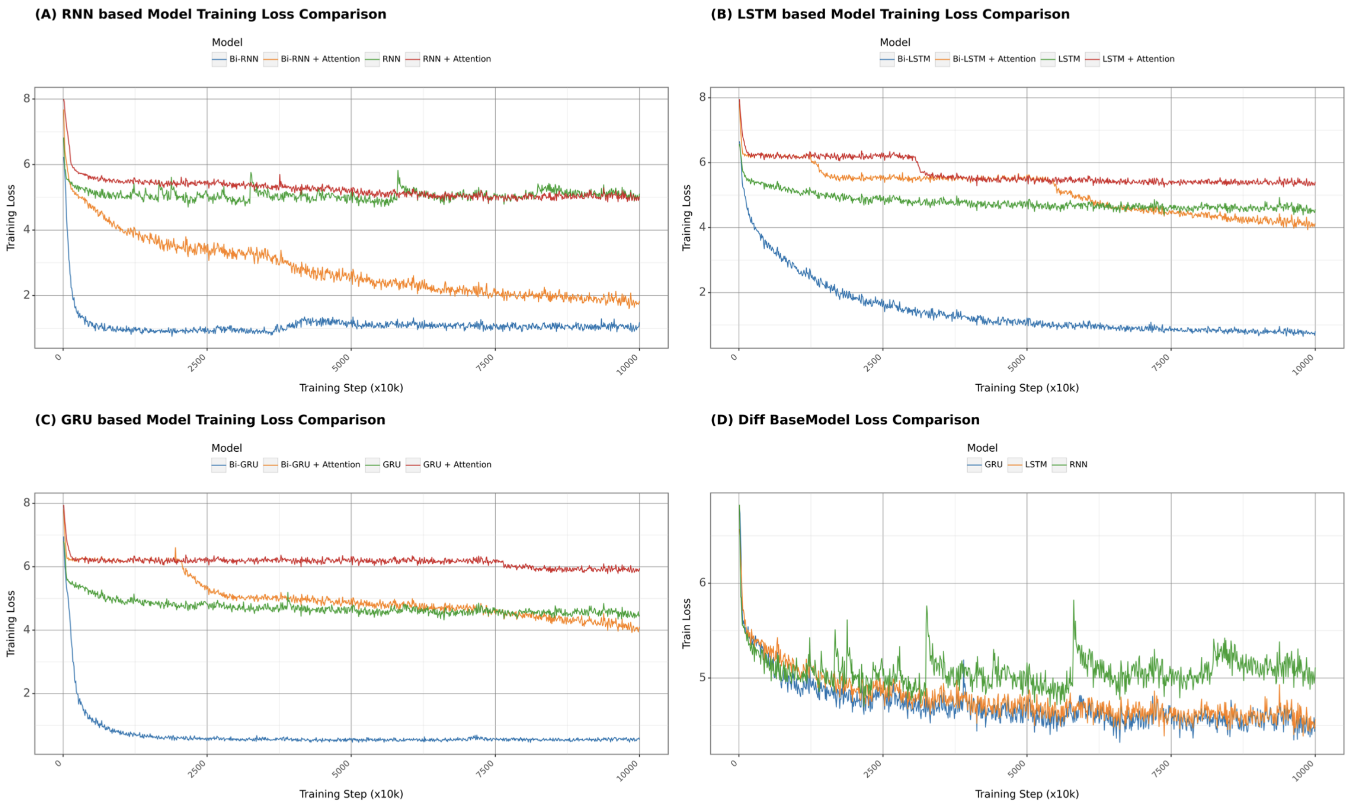

Fig. 5. Comparison of training convergence effect and final loss of models with different core modules, adding attention module and changing to bidirectional structure. (A) Comparison of three models based on RNN. (B) Comparison of three models based on LSTM. (C) Comparison of three models based on GRU. (D) Comparison of model training effects of different core modules

processing is consistent each time, thereby ensuring the stability of the model and the predictability of the generated results.

The process is shown in the fig. 4.

C. Training process analysis

The training process analysis diagram is shown in the figure.

From the training loss curve and quantitative results, we can see the performance and convergence of different models during the training process. The following is a detailed analysis of each model and its combination.

1) Comparison of Basic Models: The initial loss of the LSTM and GRU models is high, but as the training progresses, their losses decrease rapidly and converge. The final loss of the LSTM model is **6.156**, and the final loss of the GRU model is **6.147**. In contrast, the initial loss of the RNN model is relatively low, but its convergence speed is slow and the final loss is relatively high. The final loss of the RNN model is **5.820**.

2) Comparison of Bidirectional Models: The bidirectional models (Bi-LSTM, Bi-GRU, Bi-RNN) show faster convergence and lower final loss overall. The loss of Bi-LSTM and Bi-GRU decreases significantly faster than that of Bi-RNN in the early stages of training, and the final loss is also lower. The final loss of the Bi-LSTM model is **5.315**, the final loss of the Bi-GRU model is **5.157**, and the final loss of the Bi-RNN model is **4.861**.

3) Comparison of Attention Mechanisms: After adding the Attention mechanism, the initial loss of the model increased, but as training progressed, the loss dropped rapidly and reached a lower final value. The attention mechanism significantly improves the training effect of RNN, making its convergence speed and final loss comparable to LSTM and GRU. The final loss of the LSTM+Attention model is **6.013**, the final loss of the GRU+Attention model is **5.989**, and the final loss of the RNN+Attention model is **5.674**.

4) Comparison of Bidirectional + Attention Model: The bidirectional model with the Attention mechanism showed the best training effect, with a higher initial loss, but the fastest convergence speed and the lowest final loss. Bi-LSTM+Attention and Bi-GRU+Attention showed a stable loss decline trend throughout the training process, and finally reached the lowest loss value. The final loss of the Bi-LSTM+Attention model was **4.954**, the final loss of the Bi-GRU+Attention model was **4.825**, and the final loss of the Bi-RNN+Attention model was **4.492**, which was the lowest among all models.

5) Brief summary: The final loss of the 12 model training processes is shown in table III.

Compared with unidirectional models (LSTM, GRU, RNN), bidirectional models (Bi-LSTM, Bi-GRU, Bi-RNN) have obvi-

TABLE II
SELECTED PROMPT SENTENCES AND CORRESPONDING STANDARD ANSWERS

Prompt for model input	Standard answers from the dataset
寒随穷律变，春逐鸟声开。	寒随穷律变，春逐鸟声开。初风飘带柳，晚雪间花梅。
栖乌还密树，泛流归建章。	栖乌还密树，泛流归建章。华林满芳景，洛阳遍阳春。
琐除任多士，端衣竟何忧。	琐除任多士，端衣竟何忧。石鲸分玉溜，劫烬隐平沙。
日丽参差影，风传轻重香。	日丽参差影，风传轻重香。会须君子折，佩里作芬芳。
凌晨丽城去，薄暮上林栖。	凌晨丽城去，薄暮上林栖。辞枝枝暂起，停树树还低。
...	...

ous advantages in training speed and final loss. The Attention mechanism significantly improves the training effect of the model, especially the RNN model, which makes its performance close to that of LSTM and GRU. The bidirectional + Attention model shows the best training effect, the fastest convergence speed, and the lowest final loss. It is suitable as the best model choice for generating poetry. These results show that when training a poetry generation model, using a bidirectional model and adding the Attention mechanism can significantly improve the performance and training effect of the model.

TABLE III
FINAL LOSS VALUES OF DIFFERENT MODELS

Model	Final Loss
Bi-RNN+Attention	4.492
Bi-GRU+Attention	4.825
Bi-LSTM+Attention	4.954
Bi-RNN	4.861
Bi-GRU	5.157
Bi-LSTM	5.315
RNN+Attention	5.674
RNN	5.820
GRU+Attention	5.989
LSTM+Attention	6.013
GRU	6.147
LSTM	6.156

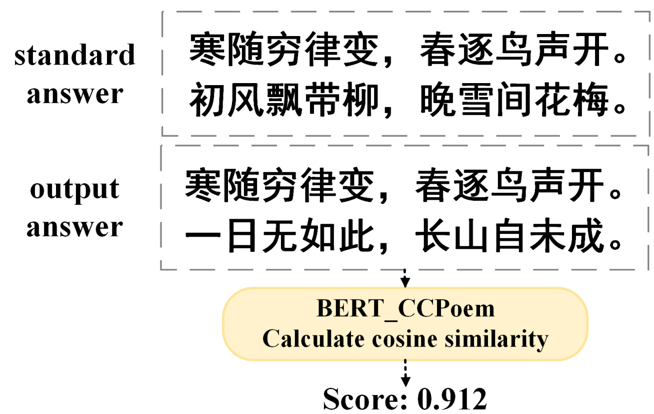

Fig. 6. Use BERT-CCPoem to calculate the model generation result score

D. Evaluation

1) BERT-CCPoem:

a) Evaluation Principles: The BERT-CCPoem model is based on the BERT architecture and is optimized for semantic understanding of Chinese classical poetry texts. BERT's bidirectional training mechanism allows the model to consider the contextual information when representing words, thereby obtaining richer and more accurate semantic information for each word. This deep bidirectional representation is necessary for understanding poetry, a genre with complex text structure and rich semantics, because the language expression of poetry usually contains multiple layers of imagery and metaphors.

b) Performance Evaluation Application: When evaluating the performance of the poetry generation model, it is implemented by calculating the semantic similarity between the generated poetry and the standard answer (the excerpt in the training set). In the experiment, I will input 5 prompts from two sentences in the dataset to the trained model. Use the BERT-CCPoem model to calculate the similarity between the two sentences generated by the model and the standard answer in the dataset, which represents the score of the performance. As shown in fig. 6. In this way, it is possible to quantify the ability of the poetry generated by the model to maintain semantic coherence and literary beauty. Using the vector representation extracted by the BERT-CCPoem model, the cosine similarity is used to measure the distance between the two text vectors, thereby evaluating the degree of closeness between the generated poetry and the reference answer at the semantic level. A high similarity indicates that the generated poem is more consistent with the standard answer in semantics, reflecting the efficiency of the model in understanding and creating ancient poetry texts.

c) Reasonableness of the evaluation method: The method of using the BERT-CCPoem model for performance evaluation is reasonable. This method can effectively capture the subtle semantic differences in the poetry text, which is crucial for evaluating whether the poetry generated by the model can achieve the language quality and artistic expression of human poetry. Since the BERT model is pre-trained on a large-

TABLE IV
EXAMPLES OF TYPICAL RESULTS OF MACHINE SCORING

Corresponding model	Generated verses	Machine score(out of 1)
LSTM	寒随穷律变，春逐鸟声开。不知无所见，一日自无人。	0.890
LSTM+Attention	寒随穷律变，春逐鸟声开。鳖锎纥纥蛄纥妹蛄纥妹纥锎蛄	0.777
LSTM+Attention	凌晨丽城去，薄暮上林栖。。。。。。。。。。。。。。。	0.902
bi-LSTM	凌晨丽城去，薄暮上林栖。栖西栖西栖西栖西栖西栖西栖西栖西	0.867
RNN	琐除任多士，端竟何忧。谁知何所远，何处不相同。	0.869
RNN+attention	寒随穷律变，春逐鸟声开。萧条虽可寞，。每勤皆可寐，况	0.880
bi-RNN	凌晨丽城去，薄暮上林栖。栖。栖。栖。栖。栖。栖。栖。栖。	0.894
bi-RNN+Attention	凌晨丽城去，薄暮上林栖。客山客山客，客绕客绕客。	0.849
GRU	寒随穷律变，春逐鸟声开。独立南山下，无人一里游。	0.916
bi-GRU+Attention	日丽参差影，风传轻重香。香生来生来生来生来生来生来生堂生	0.825

TABLE V
MANUAL SCORING STATISTICS EXAMPLE

Model	Generate results	G	C	T	Score(out of 5)	Score(out of 1)
LSTM	寒随穷律变，春逐鸟声开。不知无所见，一日自无人。	5.00	5.00	5.00	5.00	1.00
LSTM+Attention	寒随穷律变，春逐鸟声开。鳖锎纥纥蛄纥妹蛄纥妹纥锎蛄	3.00	2.00	2.00	2.60	0.52
LSTM+Attention	凌晨丽城去，薄暮上林栖。。。。。。。。。。。。。。。	1.00	1.00	0.50	0.93	0.19
bi-LSTM	凌晨丽城去，薄暮上林栖。栖西栖西栖西栖西栖西栖西栖西栖西	1.00	1.00	0.50	0.93	0.19
RNN	琐除任多士，端竟何忧。谁知何所远，何处不相同。	5.00	4.50	4.50	4.80	0.96
RNN+attention	寒随穷律变，春逐鸟声开。萧条虽可寞，。每勤皆可寐，况	4.00	3.00	4.00	3.75	0.75
bi-RNN	凌晨丽城去，薄暮上林栖。栖。栖。栖。栖。栖。栖。栖。栖。	1.50	1.00	1.00	2.30	0.26
bi-RNN+Attention	凌晨丽城去，薄暮上林栖。客山客山客，客绕客绕客。	2.00	2.00	1.50	1.93	0.39
GRU	寒随穷律变，春逐鸟声开。独立南山下，无人一里游。	2.00	1.00	0.50	1.53	0.31
bi-GRU+Attention	日丽参差影，风传轻重香。香生来生来生来生来生来生来生堂生	5.00	5.00	5.00	5.00	1.00

***Note: G represents the grammar score, C represents the expression clarity score, and T represents the coherence score.**

scale dataset, it has a wide range of semantic understanding capabilities, allowing it to accurately evaluate the similarities between poems generated by different models, which directly reflects the performance of the generative model in terms of language expression and creative expression. This evaluation strategy not only focuses on the formal beauty of poetry, but also pays more attention to its inherent semantic richness and depth of expression.

d) Selected prompt and corresponding standard answer: The 12 Chinese poetry generation models are tested on the following five prompts. The prompts and the corresponding standard answers are shown in table II.

e) Human evaluation: When evaluating the performance of the poetry generation model, a manual evaluation method was used, and three main indicators were set (all with a full score of 5): grammatical correctness, expression clarity, and thematic coherence. Grammatical correctness mainly evaluates whether the structure and word count of the poem are aligned, accounting for 0.6 of the total score. Expression clarity evaluates whether the reader can clearly understand the meaning of the poem, accounting for 0.25 of the total score. Thematic coherence checks whether the theme of the entire poem is consistent, accounting for 0.15 of the total score. These three indicators comprehensively determine the overall quality of the poem generated by the model.

The indicators and their weights are shown in table VI.

TABLE VI
POETRY GENERATION MODEL EVALUATION METRICS

Evaluation Metric	Weight
Grammatical Correctness	0.60
Clarity of Expression	0.25
Thematic Coherence	0.15

The manual evaluation score is shown in eq. (5).

$$S = 0.6G + 0.25C + 0.15T \quad (5)$$

Among in eq. (5), G represents the grammar score, C represents the expression clarity score, and T represents the the coherence score.

f) Final score: In the overall score, automated scoring and manual scoring each account for 50%.

E. Chinese poetry model test results

a) Machine test scores: Some typical output scores are shown in table IV.

From table IV, we can observe that:

1) The machine scores of all models are usually high, generally exceeding 0.85 points, which indicates that from the perspective of the model, the generated poems are well matched with the given prompts in form.
2) However, the data in Table 2 also reflect the limitations of machine scoring. Some test cases that failed to generate poems were also scored high because of their high cosine similarity.

Therefore, it can be concluded that although machine scoring can reflect a certain degree of rationality, it is not accurate enough and needs to be combined with manual scoring.

b) Manual scoring: According to the evaluation criteria, all results generated by all models are scored, and the scoring examples are statistically shown in table V.

F. Chinese poetry model ablation experiment (final score)

For Chinese poetry generation, the ablation test results of the three basic models and the corresponding improved modules (bidirectional and attention mechanism) are shown in table VII (final score: machine 50% + manual 50%).

TABLE VII
ABLATION EXPERIMENT RESULTS OF THREE BASIC MODELS AND IMPROVED MODULES (FINAL SCORES)

Core Modules	LSTM	RNN	GRU
None	0.9259	0.9690	0.9841
Attention	0.5888	0.7770	0.1770
Bidirectional	0.5130	0.2600	0.3280
Attention+Bidirectional	0.5868	0.2941	0.2600

G. Summary of performance of Chinese poetry generation model

GRU model performs best: The GRU model alone received the highest score, 98.41 points. This shows that the GRU structure is very effective in this specific poetry generation task.

RNN also performed well: The pure RNN model obtained the second highest score, 96.90 points. This also shows the powerful ability of RNN in handling text generation tasks.

LSTM's score is higher but unstable: The LSTM model itself performed well, with a score of 92.59 points. However, when the attention mechanism is added, the score drops significantly to 58.88 points.

The scores of most models dropped after adding the attention mechanism: Except for RNN, the scores of most models dropped after adding the attention mechanism. Especially for the GRU model, the score dropped from 98.41 points to 17.70 points after adding attention.

Bidirectional models perform poorly: All bidirectional models (bi-LSTM, bi-RNN, bi-GRU), regardless of whether attention is added or not, have generally low scores, mostly below 30 points. This may indicate that bidirectional structures may not be optimal in this type of generation task.

Summary: Simple GRU and RNN models show high effectiveness in this poetry generation task, while bidirectional structures and models with added attention perform poorly in most cases. This information can provide guidance for future model selection and optimization.

H. Test results on other datasets

TABLE VIII
EXAMPLES OF OUTPUT RESULTS ON OTHER DATASETS

Dataset	Output
English Articles (Shakespeare)	PELEAS: And, she hath that what's this son. If thou have to see a fool To thy mede a tongue as the since in your heard, a pates as yourselves in a marriet Thou say, the taleest of the stay on your prince ...
Chinese Lyrics (Jay Chou)	我的笑的人，不能要我不能，不用麻烦了，不能再会，我的经你的我的...
Japanese Articles	降る、これが、お夏がそれのお庇《なげ》に、このものの、その、「お夏はお夏、」「いや、そうでもいう、」という...
Codes	call_percpu(seq, sizeof(unsigned int), signime, sizeof(const), 0), { .stop_idle = strcmp, .state = pid_time, },

As shown in table VIII, some test results on other datasets are shown. After testing, the results on other datasets are basically the same as those on the Chinese poetry dataset. All three basic models can get good output results.

V. SUMMARY

This paper explores poetry generation techniques based on recurrent neural networks, especially long short-term memory networks (LSTM), basic recurrent neural networks (RNN), and gated recurrent units (GRU), by introducing attention mechanisms and bidirectional network structures. The main contributions of the study include systematically evaluating the performance of these models in poetry generation, testing the adaptability of the models on different datasets, and exploring the impact of different modules on model performance through ablation experiments.

In terms of result evaluation, the generated poems were scored using the BERT-CCPoem system to quantify the performance of the model in semantic coherence and literary beauty. Although the machine scores are generally high, indicating that the model is able to generate formally compliant poems based on the given prompts, the manual scoring results are more rigorous and better reflect the language quality and creative expression of the poems.

In terms of results, the experiments show that the GRU model performs best in the poetry generation task, followed by the RNN model, while the LSTM model, although stable without the attention mechanism, has a decline in performance after the addition. In addition, the bidirectional model and the model with the attention mechanism performed poorly in most cases, which may indicate that these complex structures do not always provide the expected results in such generation tasks.

In ablation experiments, the results show that the basic models (GRU and RNN) perform better without additional modules, which emphasizes that simplicity is sometimes more important than adding more complex structures when selecting and optimizing models.

Tests on other datasets show that these models are not only suitable for Chinese poetry generation, but also can handle different types of texts such as English articles, Chinese lyrics, Japanese articles, and codes, showing their wide applicability and flexibility.

In summary, this study not only promotes the application of machines in the field of artistic creation, especially in poetry generation, but also provides valuable insights for future model selection and structure optimization in multilingual text generation tasks. These findings emphasize the importance of seeking structure-task fit in deep learning applications.

References

[1] A. Atassi and I. E. Azami. Comparison and generation of a poem in arabic language using the lstm, bilstm and gru. *Journal of Management Information and Decision Sciences*, 25(S2):1–8, 2022.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

[3] Huimin Chen, Xiaoyuan Yi, Maosong Sun, Wenhao Li, Cheng Yang, and Zhipeng Guo. Sentiment-controllable chinese poetry generation. *arXiv preprint arXiv:2201.12345*, 2022.

[4] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[5] 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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7643–7650, Apr. 2020.

[6] Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight. Hafez: an interactive poetry generation system. In Mohit Bansal and Heng Ji, editors, *Proceedings of ACL 2017, System Demonstrations*, pages 43–48, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[7] Hugo Gonçalo Oliveira. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In Jose M. Alonso, Alberto Bugarín, and Ehud Reiter, editors, *Proceedings of the 10th International Conference on Natural Language Generation*, pages 11–20, Santiago de Compostela, Spain, September 2017. Association for Computational Linguistics.

[8] Arsh Goyal, K. Sujith, and H. R. Mamatha. Stacked lstm model for shakespeare style poem generation. In Tomonobu Senjyu, Chakchai So-In, and Amit Joshi, editors, *Smart Trends in Computing and Communications*, pages 405–415, Singapore, 2023. Springer Nature Singapore.

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[10] Bei Liu, Jianlong Fu, Makoto P. Kato, and Masatoshi Yoshikawa. Beyond narrative description: Generating poetry from images by multi-adversarial training. In *Proceedings of the 26th ACM international conference on Multimedia*, MM '18. ACM, October 2018.

[11] Dayiheng Liu, Quan Guo, Wubo Li, and Jiancheng Lv. A multi-modal chinese poetry generation model. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2018.

[12] Lixin Liu, Xiaojun Wan, and Zongming Guo. Images2poem: Generating chinese poetry from image streams. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, page 1967–1975, New York, NY, USA, 2018. Association for Computing Machinery.

[13] Yusen Liu, Dayiheng Liu, and Jiancheng Lv. Deep poetry: A chinese classical poetry generation system. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(09):13626–13627, Apr. 2020.

[14] Shakeeb A. M. Mukhtar and Pushkar S. Joglekar. Urdu, hindi poetry generation using neural networks, 2021.

[15] Alejandro Rodriguez Pascual. Bacon: Deep-learning powered ai for poetry generation with author linguistic style transfer, 2021.

[16] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, Oct 1986.

[17] M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[18] Sakib Shahriar, Noora Al Roken, and Imran Zualkernan. Classification of arabic poetry emotions using deep learning. *Computers*, 12(5), 2023.

[19] Maosong Sun, Zhipeng Guo, and Jinyi Hu. Bert-ccpoem: A pre-trained model for chinese classical poetry, 2024. Accessed: 2024-06-23.

[20] Sameerah Talafha and Banafsheh Rekabdar. Arabic poem generation with hierarchical recurrent attentional network. In *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*, pages 316–323, 2019.

[21] Tim Van de Cruys. Automatic poetry generation from prosaic text. In Dan Jurafsky, Joyce Chai, Natalie

Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2471–2480, Online, July 2020. Association for Computational Linguistics.

[22] Dongze Wu and Arbee L. P. Chen. Classical chinese poetry generation from vernacular chinese: a word-enhanced supervised approach. *Multimedia Tools and Applications*, 82(25):39139–39156, Oct 2023.

[23] Linli Xu, Liang Jiang, Chuan Qin, Zhe Wang, and Dongfang Du. How images inspire poems: Generating classical chinese poetry from images with memory networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[24] Cheng Yang, Maosong Sun, Xiaoyuan Yi, and Wenhao Li. Stylistic Chinese poetry generation via unsupervised style disentanglement. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3960–3969, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[25] Liang Yang, Zhexu Shen, Fengqing Zhou, Hongfei Lin, and Junpeng Li. Tpoet: Topic-enhanced chinese poetry generation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(6), jun 2023.

[26] Zhilin Yang, Pei Cai, Yanyan Feng, Feng Li, Wei Feng, Einat Y. Chiu, and Hong Yu. Generating classical chinese poems from vernacular chinese. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6155–6164, Nov 2019.

[27] Xiaoyuan Yi, Maosong Sun, Ruoyu Li, and Wenhao Li. Automatic poetry generation with mutual reinforcement learning. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3143–3153, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[28] Jiyuan Zhang and Dong Wang. Chinese poetry generation with flexible styles. In *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 419–423, 2018.

[29] Jianli Zhao and Hyo Jong Lee. Automatic generation and evaluation of chinese classical poetry with attention-based deep neural network. *Applied Sciences*, 12(13), 2022.

[30] Guo Zhipeng, 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 Marta R. Costa-jussà and Enrique Alfonseca, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 25–30, Florence, Italy, July 2019. Association for Computational Linguistics.